System Administrator Guide - English

FUJITSU

openFT V11.0 for z/OS

Installation and Administration

Edition August 2010

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Contents

1 Introduction

The openFT product range transfers and manages files

- automatically,
- securely, and
- cost-effectively.

The reliable and user-friendly transfer of files is an important function in a high-performance computer network. Most corporate topologies today consist of networked PC workstations, which are additionally linked to a mainframe or a Unix-based server or Windows server. This allows much of the processing power to be provided directly at the workstation, while file transfer moves the data to the mainframe for further processing there as required. In such landscapes, the locations of the individual systems may be quite far apart. Fujitsu Siemens Computers offers an extensive range of file transfer products - the openFT product range - for the following system platforms:

- BS2000/OSD®,
- Solaris™ (SPARC®/Intel™), LINUX®, AIX®, HP-UX®
- Microsoft® Windows XP™, Windows Server 2003™, Windows Vista™, Windows™ 7 and Windows Server 2008™
- OS/390 and z/OS (IBM®) respectively.

1.1 Brief description of the product openFT for z/OS

openFT for z/OS is the file transfer product for computers using the operating system z/OS.

All openFT products communicate with each other using the openFT protocol (previously known as FTNEA) as laid down by Fujitsu. Since a number of FT products from other software suppliers also support these protocols, many interconnection options are available.

With the optional FTAC function, openFT offers extended admission and access protection (FTAC stands for **F**ile **T**ransfer **A**ccess **C**ontrol).

openFT also possesses the add-on product openFT-FTP to support ftp functionality.

openFT supports the TCP/IP and SNA transport protocols.

openFT also possesses the add-on product openFT-FTP which supports ftp functionality.

1.2 Target group

This manual addresses the FT administrator and the FTAC administrator.

In order to understand the manual, knowledge of the z/OS operating system and the file transfer access methods SNA/VTAM® and TCP/IP are required.

1.3 Concept of the openFT for z/OS manuals

The openFT for z/OS product with its optional components openFT-FTP, openFT-AC and openFT-CR is described in two manuals. In addition to this System Administrator Guide, there is also a User Guide "openFT for z/OS - Managed File Transfer in the Open World”.

The manuals are arranged as follows:

- openFT for z/OS - Managed File Transfer in the Open World

  The user guide contains the following information:

  - an overview of the basic functions of the openFT product family
  - a detailed description of the conventions for file transfer to computers with different operating systems
  - a description of the user commands and the menu and program interface for the FT user
  - the openFT and openFT-AC messages for the FT user

- openFT for z/OS - Installation and Administration

  The System Administrator Guide is aimed at the FT administrator and the FTAC administrator. It describes the following:

  - how to install openFT and its optional components, including the requirements for using the product
  - how to operate, control and monitor the FT system and the FTAC environment
  - the administration commands for the FT and the FTAC administrator and also the menu and program interface
  - the openFT and openFT-AC messages for the FT administrator
  - additional sources of information for the FT administrator, such as the account records and the logging information

If openFT for z/OS is included in remote administration by means of a remote administration server, you can find information on configuring a remote administration server in the following manuals:

- “openFT V11.0 for Unix Systems - Installation and Administration” or
- “openFT V11.0 for Windows Systems - Installation and Administration”

You will also find current information and runtime application scenarios on the Internet under *ts.fujitsu.com/openft*.

1.4 Organization of the System Administrator Guide

This System Administrator Guide describes the command interface and tools available to FT and FTAC administrators. It is divided into the following chapters.

The first chapter describes the layout of this manual and the changes introduced in openFT V11.0 for z/OS as compared to the previous version V10.0.

The second chapter describes the installation of openFT V11.0 for z/OS and the prerequisites for using this product.

The third chapter describes the operation, control and monitoring of openFT and openFT-AC. It discusses how to optimize the operating parameters, the various administration activities, and what to do in the event of errors.

The fourth chapter contains the description of the menu interface for FT and FTAC administrators.

Remote administration and the associated interfaces of openFT for z/OS are introduced briefly in the fifth chapter.

The sixth chapter describes the administration commands that are used by the FT/FT-AC administrator as tools in discharging his or her administrative duties.

The seventh chapter deals with administration via the operator console.

Administration via NetView is described in chapter eight.

The appendix contains a description of the command output in CSV format, an explanation of the FT accounting records. and the openFT console messages.

1.5 Changes since the last version of the manual

The following changes have been introduced in the openFT V11.0 for z/OS System Administrator Guide since the earlier version openFT V10.0 for z/OS:

Central administration

openFT instances from different platforms can be administered using a central remote administration server that can run on a Unix or Windows system. The openFT instances can also send ADM traps (= event driven short messages) to the remote administration server on certain events.

The commands for administering the instances can also be issued from an openFT instance running under BS2000/OSD. The access rights for this instance and the address data of the openFT instances to be administered are defined in an XML-based configuration file on the remote administration server. In addition, access is controlled using special admission profiles set up on the remote administration server and the instances to be administered.

To this end, the following commands have been introduced or enhanced in BS2000/OSD:

- New command FTADM for administering the openFT instances.
- The FTMODOPT command has been expanded to allow the remote administration properties to be defined and trap logging to be controlled.
- The FTCREPRF and FTMODPRF commands have been expanded to allow the admission profiles for remote administration to be defined.
- The FTSHWLOG command has been expanded to make it possible to select on the basis of the new administration logging records and their properties.

Monitoring of openFT operation

Important monitoring data for openFT operation can be collected and output. This includes various values relating to throughput or processing time and current state indicators.

Monitoring is activated and deactivated with FTMODOPT ... ,MONITORING=*ON/*OFF.

If monitoring is activated, the monitoring data can be output using the new command FTSHWMON.

Enhanced security functions

- Secure FTP over SSL:
  Encryption is now also possible for outbound requests to a standard secure FTP server.
- 2048 bit RSA key:
  2048-bit RSA keys are now also supported.
- 256-bit AES key:
  The AES key has been expanded from 128 to 256 bit.

Prioritization of partners

Partners can be prioritized in the partner list. Requests to different partners that have the same request priority are processed in the order given by the partner priority entered.

File transfer and request queue

- More entries have been reserved in the request queue for inbound requests than in previous versions in order to guarantee as far as possible that inbound requests do not need to be rejected because there is no more space in the request queue.
- The settings regarding the file format and the protection attributes are output with the NSTATUS command.
- The FT parameter library has new parameters for the memory classes and for memory allocation.

Logging

- It is also possible to select on the basis of the request number and whether the request was for FTP when outputting log records.
- The logging ID can be up to 12 characters long.

Other changes

- Admission profiles now have a timestamp with the most recent change date. This is output with FTSHWPRF.
- A default admission profile can be set up for each user ID (with the value *STD in FTCREPRF and FTMODPRF). This profile then takes effect if the partner specifies the user ID, account and password as the transfer admission.
- openFT tracing has been optimized and the format of the trace files has been changed.

1.6 Readme file

Information on any functional changes and additions to the current product version described in this manual can be found in the product-specific Release notes. You will find them on your z/OS computer in the PO files
`<first level Qualifier>.OPENFT.FGM` and possibly
`<first level Qualifier>.OPENFTP.FPFGM` or
`<first level Qualifier>.OPENFTAC.ACFGM` or
`<first level Qualifier>.OPENFTCR.CRFGM`,
Each of these is located in the members FGMD, FGMD#FTP, FGMD#AC,or FGMD#CR for German and FGME, FGME#FTP, FGME#AC or FGME#CR for English.
The user ID under which the files are cataloged can be obtained from your system administrator. You can view the README file using the BROWSE or EDIT command and print it out on a standard printer.

2 Installation and initial operation

This chapter describes

- how to generate the data communication system for openFT,
- the general requirements that need to be observed for openFT operation (e.g. assignment of privileged for openFT and the protection of openFT administration files),
- the installation of openFT and of the optional delivery units openFT-FTP, openFT-AC and openFT-CR.
- the initial operation including the configuration and administration tasks that need to be completed before openFT is run.
- the configuration tasks associated with data security (with or without FTAC).

For information on the hardware and software requirements for openFT for z/OS and connections to partner systems, please refer to the release notice.

2.1 Generating the data communication system

Generating the data communication system for openFT comprises the following items:

- Extending the LOGON mode table and, if applicable, the COS table. This is only required if an SNA network is used for local communication or for communicating with partners.
- Generations for internal communication. Internal communication between the user commands and actual openFT processing can be performed either via VTAM or via TCP/IP (= default setting after installation) as required. To define which of these is to be used, you use the CMD_TRANS parameter in the PARM member of the openFT parameter library (see page 60). Generations are only necessary for VTAM; these are described in the following section
- Generation tasks for interconnection with partner systems via an SNA network; details are described in section “openFT interconnection via an SNA network” on page 25
- Generation tasks for interconnection with partner systems via a TCP/IP network; details are described in section “Interconnection via a TCP/IP network” on page 23

2.1.1 Extending the LOGON mode table (and, if applicable, the COS table)

The LOGON mode table of the VTAM generation must be extended by the following entry:

```
modtab   MODETAB
         MODEENT LOGMODE=FJMLMOD,                                  X
               FMPROF=X'03',                                       X
               TSPROF=X'03',                                       X
               PRIPROT=X'30',                                      X
               SECPROT=X'30',                                      X
               PSNDPAC=X'03',                                      X
               SRCVPAC=X'03',                                      X
               SSNDPAC=X'03',                                      X
               RUSIZES=X'....',                                    X
               COS=FTCOS
         MODEEND
```

The LOGMODE=FJMLMOD entry is essential.

The two macros MODETAB and MODEEND can be omitted if the MODEENT macro is inserted in an existing LOGON mode table.

The maximum lengths of the "request units" for the primary LU (first two bytes) and the secondary LU (last two bytes) specified in the RUSIZES parameter can assume values of between 1024 (X'87' for 8 x 2^7) and 32767 (e.g. X'FB' for 15 x 2^{11}). For further details, please refer to the IBM user guide for ACF/VTAM.

The specification of a "class of service table" (COS) for openFT is optional:

```
ISTSDCOS COSTAB
FTCOS    COS   VR=...
         COSEND
```

The two macros COSTAB and COSEND can be omitted if the COS macro is inserted in an existing "class of service table". For the virtual routes specified here, explicit routes must also be generated using the PATH macro.

Note that these entries - like all statements used for VTAM generation - must be entered in the correct column:

statement	starting in column 10
continuation lines	starting in column 16

2.1.2 Generations for internal communication

Only if internal openFT communication is performed via VTAM is it necessary to generate the VTAM applications for internal openFT data communication.

VTAM applications for internal openFT data communications

If internal openFT data communication is performed via VTAM then VTAM applications with the following predefined application names must be generated:

```
FJNADM (only for the STD instance)
FJNNDMS0
FJNNDMS1
  .
  .
  .
FJNNDMS9
FJAftid
FJDftid
```

Where ftid is the FT identifier. This alphanumeric character string may consist of a maximum of five characters and must be unique for all linked FT systems. This ftid must also be specified in the FJGEN command (see page 182) for the openFT instance. The entries FJAftid and FJDftid must exist for each openFT instance that uses internal communication via VTAM. These are the instances for which CMD_TRANS=TCP is **not** set in the PARM member of the parameter library.

The VTAM applications FJNADM, FJNNDMS0,... , FJAftid and FJDftid are used for internal openFT communication. FJNADM is only used by the instance STD. As a minimum requirement, you must generate the applications FJNADM and FJNNDMS0. Up to 10 applications (in continuous ascending order from FJNNDMS0 to FJNNDMS9) can be generated. This provides users with more connections for the entry of the commands (this also applies to the program interface and menu interface).

These VTAM applications are generated using the following statements:

```
         VBUILD   TYPE=APPL
FJNADM   APPL     ACBNAME=FJNADM,                         X
                  AUTH=(ACQ,VPACE),                       X
                  DLOGMOD=FJMLMOD,                        X
                  MODETAB=modtab,                         X
                  PRTCT=ft-password,                      X
                  VPACING=3
FJNNDMS0 APPL     ACBNAME=FJNNDMS0,                       X
                  AUTH=(ACQ,VPACE),                       X
                  DLOGMOD=FJMLMOD,                        X
                  MODETAB=modtab,                         X
                  VPACING=3
```

```
FJNNDMS1 APPL      ACBNAME=FJNNDMS1,                                       X
                   AUTH=(ACQ,VPACE),                                       X
                   DLOGMOD=FJMLMOD,                                        X
                   MODETAB=modtab,                                         X
                   VPACING=3
FJNNDMS2 APPL      ACBNAME=FJNNDMS2,                                       X
                   AUTH=(ACQ,VPACE),                                       X
                   DLOGMOD=FJMLMOD,                                        X
                   MODETAB=modtab,                                         X
                   VPACING=3
.
.
.
FJAftid  APPL      ACBNAME=FJAftid,                                        X
                   AUTH=(ACQ,VPACE),                                       X
                   DLOGMOD=FJMLMOD,                                        X
                   MODETAB=modtab,                                         X
                   PRTCT=ft-password,                                      X
                   VPACING=3
FJDftid  APPL      ACBNAME=FJDftid,                                        X
                   AUTH=(ACQ,VPACE),                                       X
                   DLOGMOD=FJMLMOD,                                        X
                   MODETAB=modtab,                                         X
                   PRTCT=ft-password,                                      X
                   VPACING=3
```

where

modtab
: is the name of the LOGON mode table (see page 20),

ft-password
: is the FT password which can be used to protect the VTAM applications, the request file, the partner list and the trace files of an openFT installation. The password is specified in the installation parameters (see page 49) or in the FT administration command FJGEN (see page 182). The applications FJNNDMS0 ... FJNNDMS9 must not be protected by an FT password. For all other applications, password protection is optional.

The entry DLOGMOD=FJMLMOD is essential in these statements.

2.1.3 Interconnection via a TCP/IP network

The following generation operations must be performed in order to connect to FT partner systems via a TCP/IP network:

- The connection between openFT for z/OS and the software product TCP/IP for MVS must be generated.
- The transport system address of the local openFT- instance must be determined.
- The transport system addresses of the remote partner systems must be entered.

2.1.3.1 Transport system address of the local openFT instance

In the case of a TCP/IP interconnection, the transport system address of a local openFT instance consists of the Internet address, port number and T-selector.

The **Internet address** of the z/OS system on which the local openFT instance is running is assigned using the HOST NAME parameter in the FJGEN command (see page 182). In HOST NAME, you should always directly specify the IP address or the host name. If multiple openFT instances are to be able to run in parallel on the z/OS system then they must be assigned different IP addresses. Please note that you may only use IP addresses that are defined in your z/OS system’s address space.

The **port number** of the openFT for z/OS main station (passive port) is defined using the OPENFT-APPL parameter in the FTMODOPT command (see page 343). We strongly recommend you use the default port number 1100. This is predefined as the default in all openFT products and therefore greatly simplifies addressing in a heterogeneous environment.

The **T-selector** of the openFT for z/OS main station has the name $FJAM␣␣␣.

2.1.3.2 Transport system addresses for TCP/IP partner systems

The transport system address of a TCP/IP partner system consists of the Internet address of the remote computer, the T-selector of the remote FT system and, where appropriate, the port number of the RFC1006 implementation of the remote FT system if this differs from the default port number 1100. The transport system addresses of all partner systems which are to be accessed via TCP/IP must be reported to openFT for z/OS. This has been considerably simplified as of openFT V10:

- The FT administrator enters the partner systems in the partner list with the FTADDPTN command (see page 195) and stores the necessary address information there. For further details on specifying addresses, see page 108.

  If host names are used, it must be possible to determine the associated IP address from the relevant data source, e.g. from the file TCPIP.HOSTS.LOCAL or using DNS.

- If dynamic partners are permitted (see page 112), it is also possible to directly address a partner system of this type without it being entered in the partner list.

Entries in the TNSTCPIP member of the FT parameter library are no longer necessary. Existing entries can, however, still be used, see section “Structure of the member TNSTCPIP” on page 77.

2.1.4 openFT interconnection via an SNA network

Further VTAM applications must be generated for interconnection with FT partner systems via an SNA network, irrespective of the type of partner system in question (openFT for z/OS, openFT for Windows and the connection method of Microsoft's Host Integration Server used there):

- a main station that receives all external transfer requests ("inbound submissions")
- substations that are used for transfer requests initiated by the local openFT instance ("outbound submissions").

Naming conventions

The names of these VTAM applications are formed in accordance with a set of naming conventions. They start with a prefix (main station: FJM..., substations: A01..., A02... etc.) to which the ftid of the local openFT instance is added (see section "Generations for internal communication" on page 21).

This convention makes it possible to assign unique names to VTAM applications. These names are then used by the local openFT instance when establishing the SNA connection to remote systems.

It also provides unique identification of all interconnected partner systems at transport system level and therefore facilitates extended authentication (see page 118). If extended authentication is enabled in a remote FT system then, in the case of an SNA connection, the remote system will only accept inbound requests if it is able to assign the substation name (A01ftid, A02ftid etc.) that appears as the sender address to the name of the partner system's main station entered in the partner list (FJMftid). You can activate extended authentication in z/OS by using PARTNER-CHECK=*TRANSPORT-ADDRESS in the command FTMODOPT, see page 254). It is only possible to deviate from these main station and substation names ("free VTAM names"), which apply throughout the SNA network, if extended authentication is not used. For further information, refer to the notes below.

The following VTAM applications must therefore be generated:

```
FJMftid
A01ftid
A02ftid
   .
   .
   .
Annftid
```

where:

ftid
FT identifier. This alphanumeric character string can consist of up to five characters and must be unique among all interconnected FT systems.

nn
is greater than or equal to the maximum number of transport connections (defined with the operating parameter CONNECTION-LIMIT, see page 257). These numbers (01, 02,... nn) must be assigned in continuous ascending order as otherwise not all the generated VTAM applications can be used by openFT.

These VTAM applications are generated using the following statements:

```
FJMftid  APPL      ACBNAME=FJMftid,                             X
                   AUTH=(ACQ,VPACE),                            X
                   DLOGMOD=FJMLMOD,                             X
                   MODETAB=modtab,                              X
                   PARSESS=YES                                  X
                   PRTCT=ft-password,                           X
                   VPACING=3
A01ftid  APPL      ACBNAME=A01ftid,                             X
                   AUTH=(ACQ,VPACE),                            X
                   DLOGMOD=FJMLMOD,                             X
                   MODETAB=modtab,                              X
                   PARSESS=YES                                  X
                   PRTCT=ft-password,                           X
                   VPACING=3
A02ftid  APPL      ACBNAME=A02ftid,                             X
                   AUTH=(ACQ,VPACE),                            X
                   DLOGMOD=FJMLMOD,                             X
                   MODETAB=modtab,                              X
                   PRTCT=ft-password,                           X
                   VPACING=3
.
.
.
Annftid  APPL      ACBNAME=Annftid,                             X
                   AUTH=(ACQ,VPACE),                            X
                   DLOGMOD=FJMLMOD,                             X
                   MODETAB=modtab,                              X
                   PRTCT=ft-password,                           X
                   VPACING=3
```

where:

modtab
is the name of the LOGON mode table (see page 20),

ft-password
is the FT password which may be used to protect the VTAM applications, the request file, the partner list and the trace files of an openFT installation. The password is specified in the FT parameter library (see page 49) or in the FT administration command FJGEN (see page 182). This password specification is optional for the VTAM applications described here.

The entry DLOGMOD=FJMLMOD is essential in these statements.

Notes

- During VTAM or NCP generation, it is also necessary to enter the main station and substations of all FT partner systems which are connected via SNA. However, these entries depend on the type of partner system in question (openFT for z/OS, openFT for Windows) and on the connection method used (Host Integration Server from Microsoft). These entries are therefore described in the sections dealing with the individual partner systems (page 29).
- Entries in partner systems using openFT V10 and higher: The name of the main station (FJMftid) of the local openFT instance can be specified in the partner list of the remote FT system (e.g. in the PARTNER-ADDRESS operand of the FTADDPTN command). Examples of the interrelation of VTAM generation and the partner list entries can also be found in the sections dealing with the individual partner systems (page 29).
- Entries in partner systems using openFT < V10: The name of the main station (FJMftid) of the local openFT instance must also be specified in the network description of the remote FT system, e.g. in the NETWORK-ADDRESS operand of the FTADDPTN command see page 195.
- You are advised to generate at least 8 applications A01..., A02..., A03... etc. in order to avoid bottlenecks. The substations of the local openFT instance, A01ftid to Annftid, must be numbered in continuous ascending order, otherwise not all the applications and transport connections available can be used.
- In general, the names formed in accordance with the naming conventions described above must be entered both as the name of the VTAM application and as the value of the ACBNAME name (see APPL statements above). Only then can the remote FT system perform extended authentication (see page 118) as specified in case of a remote openFT for z/OS in the PARTNER-CHECK=*TRANSPORT-ADDRESS parameter of the FTMODPTN command (see page 254).

However, if "free" main station and substation names (which apply throughout the SNA network) are to be used, then the following applies:

- The "free name" is specified as the name in the APPL statement. However, the name formed in accordance with the naming conventions described above must still be specified for the ACBNAME parameter.
- The remote FT systems must/can contain corresponding entries in the network description file/partner list. The entry is mandatory in systems using openFT < V10 and is optional in systems using openFT V10 and higher. If the remote FT system is also an openFT for z/OS, the "free VTAM name" of the local openFT instance's main station must/can also be specified in the FTADDPTN command (as a value for the NETWORK-ADDRESS parameter *in* openFT < V10 or, as of V10, as a value of the PARTNER-ADDRESS parameter). If openFT for z/OS V6 or earlier is used in the remote system then the "free VTAM name" of the local system's main station must be specified there as a value for the RELADR parameter in the FJADDSYS command. The name of the local system's main station formed in accordance with the naming conventions must still be specified as SYSADR, i.e. FJMftid.
- If a connection is made to openFT for z/OS or to openFT V8.1 for Windows with Microsoft's Host Integration Server, then the corresponding entries in the conversion tables or name servers of these products must be made.
- If a "VTAM Interpret Table" is generated in the local system, it must also contain the names of the openFT VTAM applications

  ```
  FJMftid
  A01ftid
  A02ftid
  etc.
  ```

- If "free VTAM names" are used for the stations of the local openFT instance, none of the partner systems to which this system is connected may operate extended authentication.

The following sections deal with the individual partner systems and contain examples of "free VTAM names".

"Free VTAM names" can only be used for main stations and substations (FJMftid, Annftid). They cannot be used for VTAM applications for internal openFT data communications (FJNADM, FJNNDMS0 ... FJNNDMS9, FJAftid, FJDftid)

2.1.4.1 openFT interconnection of two z/OS systems via an SNA network

No extensions to the VTAM generation beyond the entries necessary for the homogeneous interconnection of two z/OS systems and the extensions to the LOGON mode table and the VTAM applications described above are required when the two openFT for z/OS systems are to be interconnected via an SNA network.

Since the interconnected systems are located in different domains, the VTAM applications used for data transfer (FJMftid, A01ftid, A02ftid, etc.) must be defined as "cross domain resources".

Example

Linking two systems openFT for z/OS:
FTZOS1 (ftid = *ZOS1*) and *FTZOS2* (ftid = *ZOS2*).

- The VTAM applications FJNADM, FJNNDMS0 (at least), FJAZOS1, FJDZOS1, FJMZOS1, A01ZOS1, ... , A08ZOS1 must be generated in VTAM on the z/OS computer with the FT system *FTZOS1* (specify PARSESS=YES with FJMZOS1, A01ZOS1).
- The VTAM applications FJNADM, FJNNDMS0 (at least), FJAZOS2, FJDZOS2, FJMZOS2, A01ZOS2, ... , A08ZOS must be generated in VTAM on the z/OS computer with the FT system *FTZOS2* (specify PARSESS=YES with FJMZOS2, A01ZOS2).
- The VTAM applications used for transport (FJMZOS1, A01ZOS1, ..., A08ZOS1, FJMZOS2, A01ZOS2, ... , A08ZOS2) must additionally be defined as "cross domain resources".
- In addition, the LOGON mode table for openFT must be generated in both computers.
- The remote FT system *FTZOS2* can be entered as follows in the partner list of the FT system *FTZOS1*:

  ```
  FTADDPTN PARTNER-NAME=FTZOS2,PARTNER-ADDRESS=FJMZOS2:SNA
  ,IDENTIFICATION= ...)
  ```

  If you address FTZOS2 directly in FT requests then specify FJMZOS2:SNA.
- The remote FT system *FTZOS1* can be entered as follows in the partner list of the FT system *FTZOS2*:

  ```
  FTADDPTN PARTNER-NAME=FTZOS1,PARTNER-ADDRESS=FJMZOS1:SNA
  ,IDENTIFICATION= ...)
  ```

  If you address FTZOS1 directly in FT requests then specify FJMZOS1:SNA.
- In each case, the instance ID of the partner system in the IDENTIFICATION parameter of the FTADDPTN command must be entered in the same way as it has been defined there in the FTMODOPT command.

The same example using "free VTAM names"

The following names, which apply throughout the network, should be used for the *FTZOS1* FT system:

MVSMAIN	(for FJMZOS1)
MVSSUB1	(for A01ZOS1)
MVSSUB2	(for A02ZOS1)
...	...
MVSSUB8	(for A08ZOS1)

The VTAM-APPL statements for these applications on the computer with the FT system *FTZOS1* are as follows:

```
MVSMAIN  APPL   ACBNAME=FJMZOS1, ...
MVSSUB1  APPL   ACBNAME=A01ZOS1, ...
MVSSUB2  APPL   ACBNAME=A02ZOS1, ...
...
MVSSUB8  APPL   ACBNAME=A08ZOS1, ...
```

The remote openFT system ***FTZOS1*** can be entered as follows in the partner list of the openFT system ***FTZOS2***:

```
FTADDPTN PARTNER-NAME=FTZOS1,PARTNER-ADDRESS=MVSMAIN:SNA
,IDENTIFICATION='ZOS1.FUSINET.AT'
```

2.2 Installation of openFT

2.2.1 Preparations for installation

Before reading in the product tape, you should make a few preparations, such as defining the admissions for the openFT user IDs and the openFT privileges and regulating protection of the administration files.

2.2.1.1 User IDs for openFT

The following user IDs are required for openFT operation:

– a user ID under which openFT runs (as a job or started task, see page 85)

– one or more FT administrator IDs

– one or more FTAC administrator IDs (only if the FTAC functionality is used)

openFT can run under an ID without TSO authorization, if this is required on account of the privilege level which this user ID needs (see next section).

The user IDs that openFT or FTAC can administer are defined in the FTADM and FTACADM members of the FT parameter library PARM (see page 62).

The internal data sets that are required to operate the openFT instances are catalogued with the prefix OPENFT QUALIFIER. In particular, this applies to the instance’s request file and the partner list (see section "Internal openFT data sets" on page 424). The OPENFT QUALIFIER is specified in the FJGEN command (see page 48 and page 182).

2.2.1.2 openFT privileges

When installing openFT it is important to note the following points concerning privileges:

● If the product RACF (or compatible product) is installed in the system, the OPENFT load module must be stored in a library which is subject to APF authorization, since it accesses privileged RACF macros (see the section “Linking openFT with data protection products” on page 90). In addition, the OPENFT load module must possess the linkage editor attribute "AC(1)". The OPENFT load module supplied already has this attribute.

  openFT must also have APF authorization in order to perform the following functions:

  – transfer a complete PO or PDSE data set
  – charge file transfer requests (write account records to the SMF file)
  – output asynchronous messages after termination of a transfer request to the TSO user whose user ID was specified in the TRANSFER-ADMISSION of the system involved and/or to one or several consoles.

In addition to the library containing the OPENFT load module, the other libraries of the library hierarchy STEPLIB, TASKLIB, JOBLIB ... APF must also be authorized, i.e.:

- the library containing openFT as a subsystem, known as the LPALIB
- the library containing the OPENFTCR load module (see section “Installation of the openFT-CR delivery unit” on page 42)

- Since openFT uses socket calls to establish TCP/IP connections, the user ID under which openFT runs (as a job or as a started task, see page 85) also needs an OMVS segment (OMVS: OpenEdition MVS). No special privileges are needed, i.e. any UID (OMVS user ID) can be used. The user ID must belong to a group for which a GID (OMVS group ID) has been defined. The GID is defined with RACF; see also IBM manual "OpenEdition Planning", chapter "Controlling OpenEdition Security".
- If the file SYS1.UADS is installed in the system and is to be used by openFT, the user ID under which openFT is running (as a job or started task, see page 85) must be granted read access to this file.
- In an z/OS system with RACF (Resource Access Control Facility), the user ID under which openFT is running must also be authorized to access the files and volumes of all openFT users if these are protected by RACF. In particular it must be granted:

  - read access (READ) to send files
  - write access (ALTER) to receive files

  The z/OS administrator can assign specific access rights to these files and to the associated data volumes. However, it is considerably easier to assign the RACF attribute OPERATIONS to the user ID under which openFT is running. If this approach is taken, it is advisable to not to assign any TSO authorization to this user ID for reasons of data security. Even if the user ID under which openFT is running possesses the RACF OPERATIONS attribute and is therefore able to access all the files in the system, there is no danger of FT user transfer requests infringing on data security, since openFT verifies the validity of all the data access attempts that occur during file transfer (see section “Linking openFT with data protection products” on page 90).

  The same rules apply to products compatible with RACF. For further information please refer to the product-specific manuals.

2.2.1.3 Protecting openFT administrative files

The data sets created for the administration and operation of openFT should be protected against unauthorized access (e.g. by using RACF). The degree of protection needed will vary depending on the particular security requirements of individual computer centers. The following sections contain recommendations for protecting the most important data sets. For some of the data sets, the most stringent access restrictions that will still allow openFT operation are described.

FT parameter library

The parameters with which openFT is adjusted to installation-specific requirements (see section “Setting up the FT parameter library” on page 49) are stored in the FT parameter library. This is highly sensitive information, the integrity of which is absolutely essential for openFT to be able to function properly (for instance the list of FT or FTAC administrators and possibly the name the FTAC file; see below). This file must therefore be protected extremely carefully.

Request file, partner list, operational parameters file

The request queue, the partner list and the operational parameters file are three DA data sets set up automatically under the following names the first time the system is started:

- The request queue '<openft qualifier>.<inst>.*SYSRQF*'
- The partner list '<openft qualifier>.<inst>.*SYSPTF*'
  The partner list contains the address information for the partner systems and corresponds to the network description file used in previous openFT versions.
- The operational parameters file '<openft qualifier>.<inst>.*SYSOPF*'.

Here, <openft qualifier> is the prefix with which the openFT administrative files are created (OPENFT QUALIFIER in the FJGEN command). <inst> is the instance name (INSTANCE NAME in the FJGEN command).

These three files only need to be accessed by the user ID under which openFT is running.

Logging file

The logging file is generated automatically by openFT. Its components are described in section “Internal openFT data sets” on page 424.

The names of the components of the logging file all begin with ’<openft qualifier>.<inst>.SYSLOG'. "openft qualifier" is the prefix with which the openFT administrative files are created (OPENFT QUALIFIER in the FJGEN command). "inst" is the instance name (INSTANCE NAME in the FJGEN command). Instead of the usual second level qualifier inst.SYSLOG, the administrator may allocate a different name to the file (LOGFILE_2ND_Q key in the PARM member of the FT parameter library).

Beside the main FT administrator ID, only the user ID under which openFT is running should be able to access the components of the logging file. (Please also read the note at the end of section "FTAC files".)

If you want to store the logging records permanently, redirect the output from the FTSHWLOG command to a file and then back up this file. To prevent the logging file from becoming unnecessarily large, you should occasionally use the FTDELLOG command to delete old logging records.

FTAC file

The FTAC file is generated automatically by openFT when FTAC is used. It contains the FTAC environment, i.e. the admission sets, admission profiles, etc. The components of the file are described in section “Internal openFT data sets” on page 424.

The names of the components of the FTAC file all begin with '<openft qualifier>.<inst>.SYSFSA'. "openft qualifier" is the prefix with which the openFT administrative files are created (OPENFT QUALIFIER in the FJGEN command). "inst" is the instance name (INSTANCE NAME in the FJGEN command) Instead of the usual second level qualifier inst.SYSFSA, the administrator may allocate a different name to the file (FILE_2ND_Q key in the FTACPAR member of the FT parameter library).

For reasons of security it is strongly recommended that the components of this file be accessible only to the main FT administrator ID and the user ID under which openFT runs.

Note

> If you are using RACF and you want to protect the logging file and the FTAC file using generic profiles, you must make sure that all components of the files are covered by the names of the generic profiles.
>
> If you want to use to implement a standard protection for the request file, the partner list, the logging file and the FTAC file and if you select the same beginning for the file names of all of these files then you will need only two generic profiles to protect them.

If you use the standard file names for the files, you only need to implement the following generic profiles for the individual openFT instances:

’<openft qualifier>.<inst>.SYS*’

This generic profile protects the request file (SYSFSF), the partner list (SYSPTF) and the PS data sets that are part of the logging file and the FTAC file (SYSLOG and SYSFSA).

’openft qualifier.inst.SYS*.*’

This generic profile protects the components of the VSAM cluster, which are part of the logging file and the FTAC file (SYSLOG.P00 etc. for the logging file, SYSFSA.P00 etc. for the FTAC file).

The OPENFT QUALIFIER stands for the file name prefix defined in the FJGEN command, while inst refers to the instance name defined for the corresponding openFT instance in the INSTANCE NAME parameter in the FJGEN command.

2.2.2 Reading in the openFT product tape

openFT for z/OS is supplied for installation with SMP/E (System Modification Program/Extended) as a "Custom-Built Product Delivery Offering" (CBPDO) as "function SYSMOD" with the following characteristics:

File name prefix (RFDSNPFX):	OPENFT
Identification (FMID):	OFT110A

In order to install openFT, an SMP/E environment for openFT has to be created; amongst other things, this comprises a product-specific "Consolidated Software Inventory" (CSI). A set of procedures is supplied with which an SMP/E environment for openFT is created and with which the SMP/E statements RECEIVE, APPLY and ACCEPT are executed.

i For installation, the product data is also offered on CD as well as product tape. You will find further information and installation instructions on the CD.

2.2.2.1 Contents of the product tape

The product tape, which contains the openFT data sets, is created as an "SL tape" using the IBM utility IEBCOPY. It contains the following files in the order in which they are listed below:

OPENFT.OFT110A.SMPMCS
MCS statements for SMP/E (MCS: Modification Control Statement)

OPENFT.OFT110A.F1
PO data set containing the following CLIST's:

OFT110A	JCLIN statements for transferring the other product modules from the tape with SMP/E (JCLIN: Job Control Input)
OPFT#01	creates the system and backup files for openFT
OPFT#02	installs the SMP/E environment for openFT
OPFT#03	initializes the SMP/E environment for openFT
OPFT#04	executes the SMP/E statement RECEIVE
OPFT#05	executes the SMP/E statement APPLY
OPFT#06	executes the SMP/E statement ACCEPT

OPENFT.OFT110A.F2
PO data set containing the following members:

FGMD	contains the Release Notice in printable form in German
FGME	contains the Release Notice in printable form in English

OPENFT.OFT110A.F3
PO data set containing samples for FT users and FT administrators (the $$INDEX member contains a brief description of the other members).

OPENFT.OFT110A.F4
FT basic procedure library with the CLIST FJGEN. The other CLIST procedures are stored in instance-specific FT procedure libraries during the FJGEN run.

OPENFT.OFT110A.F5
openFT Load module library containing the following members:

OPENFT	Program openFT **without** SSL encryption
OPENFTS	Alias name for OPENFT (see Explanation (1) below)
OPENFTSL	Program openFT **with** SSL encryption
OPENFTSS	Alias name for OPENFTSL (see Explanation (1) below)
OPFTSUBL	Subsystem handler

(1) The aliases OPENFTS and OPENFTSS are identical to the entries OPENFT and OPENFTSL in terms of their functions. The alias can be used in place of this entry if it is necessary to refer to the load module using a name other than the user ID under which the openFT job is running.

OPENFT can also be used for console applications or NetView applications.

NCOPY.OFT110A.F6
Load module library containing the members
FTADDPTN, FTADM, FTMODREQ, FTMODOPT, FTMODPTN, FTREMPTN, FTSHWMON, FTSHWOPT, FTSHWPTN, FTSHWD, FTSTART, FTSTOP, FTCREKEY, FTDELKEY, FTUPDKEY, FTSHWLOG, FTSHWNET, FTSHWINS, FTUPDPAR, FTDELLOG, FTHELP, FTTERM, NCOPY, NSTATUS, NSTAT, NCANCEL, NCAN, FTSCOPY, FTACOPY, FTCANREQ, FTSHWREQ, FTDEL, FTMOD, FTSHW, FTCREDIR, FTMODDIR, FTDELDIR, FTUPDKEY, FTEXEC, FTTRACE, OPFTSUBL.
The following alias names can only be used if openFT-AC is installed
FTCREPRF, FTDELPRF, FTMODADS, FTMODPRF, FTSHWADS, FTSHWPRF, FTSHWRGE, FTEXPENV, FTIMPENV, FTSHWENV.

OPENFT.OFT110A.F7
Macro library containing the ASSEMBLER macro OPENFT.

OPENFT.OFT110A.F8
Library containing the ISPF panel definitions for the menu interface for FT users and FT administrators

OPENFT.OFT110A.F9
Library containing the corresponding CLIST procedures.

OPENFT.OFT110A.F10
Library containing the corresponding message definitions.

OPENFT.OFT110A.F11
Contains the members IGX00211, OPFTIGX, OPFTINIT and OPFTSUB. These objects are installed in a separate PO library with the name LPALIB during installation. Users can then copy them to any location (e.g. to SYS1.OPENFT.LPALIB) in order to then start openFT as a subsystem.

OPENFT.OFT110A.F12
Contains the code tables IBM037, IBM273 and IBM500.

2.2.2.2 How to proceed with the installation

Proceed as follows to install openFT:

1. Copy the second file (OPENFT.OFT110A.F1) from the product tape to a PO data set using IEBCOPY (a sample job is shown below).
2. Adjust the CLISTs in members OPFT#01 to OPFT#06 of the copied PO data set to the computer center. You can, for example use the T_BASE variable to assign the installation location for openFT. By default, openFT is installed at the location to which the files were copied for installation purposes.
3. Execute the modified CLISTs:

OPFT#01	creates the system and backup files for openFT
OPFT#02	installs the SMP/E environment for openFT
OPFT#03	initializes the SMP/E environment for openFT
OPFT#04	executes the SMP/E statement RECEIVE
OPFT#05	executes the SMP/E statement APPLY
OPFT#06	executes the SMP/E statement ACCEPT

If the optional delivery units openFT-CR (see page 42), openFT-AC (see page 44) or openFT-FTP (see page 46) are not to be installed, you can now continue with the steps described in section “Making the commands and the ISPF panels available” on page 40, and section “Setting openFT installation parameters with FJGEN” on page 48 etc.

Sample job for copying the file OPENFT.OFT110A.F1

```
//USERA#   JOB (ACCOUNTING),MSGCLASS=X,CLASS=A,
//             REGION=0M,NOTIFY=USERA,USER=USERA,PASSWORD=
//*****************************************************************
//**                                                            **
//**            READ IN THE openFT PRODUCT TAPE                 **
//**            SECOND DATA SET ONLY (OPENFT.OFT110A.F1)        **
//**                                                            **
//*****************************************************************
//*
//IEBCOPY   EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//INDD1    DD  DISP=SHR,LABEL=(2,SL),DSN=OPENFT.OFT110A.F1,
//             UNIT=TAPE,VOL=SER=(FT100A)
//OUTDD1   DD  DSN=USERA.OPENFT.JCLIN,
//             DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,VOL=SER=654321,
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//             SPACE=(TRK,(10,2,1),RLSE)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DISP=(,DELETE)
//SYSUT4   DD  UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DISP=(,DELETE)
//SYSIN    DD  *
  COPY O=OUTDD1,I=INDD1
/*
//
```

Note

If openFT is to run with APF-authorization, then the following libraries must have APF authorization:

- the library that contains the OPENFT or OPENFTSL load module (i.e. the library to which the OPENFT.OFT110A.F5 library was copied from the product tape)
- and the library that contains openFT as a subsystem (i.e. the library to which the library OPENFT.OFT110A.F11 is copied from the product tape).

This also applies to the other libraries in the library hierarchy STEPLIB, TASKLIB, JOBLIB ... (see section “openFT privileges” on page 31).

2.2.3 Making the commands and the ISPF panels available

When the openFT product tape has been read in, you must make sure that the library containing the openFT commands (OPENFT.LOAD and OPENFT.NCLOAD) and the libraries containing the ISPF panels, CLISTs and messages for the FT administrator menu interface are available to the user IDs which are authorized to use these commands or this menu interface. In other words, you must concatenate the libraries accordingly.

2.2.3.1 Concatenating libraries with the openFT commands

This section contains a description of two different ways of making the CLIST libraries containing the administration commands (FT procedure library) available to those user IDs that are authorized to use these commands.

Modifying the LOGON procedure

The LOGON procedure is executed each time a user logs on. During this procedure the FT procedure library is concatenated with the library containing the defined system procedures ('system-procedure-library'):

```
//SYSPROC  DD  DSN=system-procedure-library,DISP=SHR
//         DD  DSN=ft-procedure-library,DISP=SHR
//OPENFT   DD  DSN=<openft qualifier>.<inst>.CONN,DISP=SHR
```

If you want to use the menu interface for the FT administrator, you must make some further modifications to the LOGON procedure (see section “Concatenating libraries with the openFT commands” on page 40**).**

Making the library available dynamically in a TSO session

The library can alternatively be made available during a TSO session by means of the following TSO commands:

```
ALLOC FILE(SYSPROC) DSNAME('system-procedure-library' -
'ft-procedure-library') REUSE SHR
```

and the two commands:

```
ALLOC DSNAME('<openft qualifier>.<inst>.CONN') DDNAME(OPENFT) SHR REUSE
```

```
TSOLIB ACT DATASET('<openft qualifier>.OPENFT.NCLOAD')
```

Following this, the FT commands are available during the current session.

You are advised to protect the library containing these CLISTs, e.g. by means of RACF, in order to protect these procedures and thus the entire FT administration against unauthorized access.

2.2.3.2 Concatenating libraries containing ISPF panels

The ISPF panel definitions and the associated MSG and CLIST libraries are contained in three PO data sets:

```
OPENFT.OFT110A.F8   contains the panel definitions
OPENFT.OFT110A.F9   contains the CLIST procedures required for execution
OPENFT.OFT110A.F10   contains the message definitions
```

These libraries, into which the components were read from the openFT product tape (see section “Reading in the openFT product tape” on page 36), must be made accessible to those user IDs who are authorized to administer openFT via the menu interface, using the methods already described above. This description deals only with the modification of the LOGON procedure; the information given above on making the library available during a TSO session also applies here.

Modification of the LOGON procedure

```
//SYSPROC  DD  DSN=system-procedure-library,DISP=SHR
//         DD  DSN=ft-procedure-library,DISP=SHR
//         DD  DSN=ft-clist-library,DISP=SHR
//OPENFT   DD  DSN=<openft qualifier>.<inst>.CONN,DISP=SHR
//ISPPLIB  DD  DSN=system-panel-library,DISP=SHR
//         DD  DSN=ft-panel-library,DISP=SHR
//ISPMLIB  DD  DSN=system-message-library,DISP=SHR
//         DD  DSN=ft-message-library,DISP=SHR
```

The "ft-procedure-library" is the library described above for the FT administration commands. "ft-clist-library" is the above-mentioned library for the CLIST procedures in the openFT menu interface (after installation, OPENFT.PANEL.CLIST), "ft-panel-library" is the library for panel definitions (after installation, OPENFT.PANELS) and "ft-message-library" is the library for messages (after installation, OPENFT.PANEL.MSG). These libraries, too, should be protected against unauthorized access, e.g. using RACF.

CLIST command procedures for the FT administration commands are created when the FJGEN procedure is executed (see page 48) and stored in the library <openft qualifier>.<inst>.CLIST.

Starting the panel interface

Call the following command under TSO:

```
EXECUTE '<openft qualifier>.OPENFT.PANEL.CLIST(FJMENU)'
```

By calling this start CLIST, you can access the initial panel of the openFT menu interface both under TSO and via the general ISPF/PDF interface (enter "TSO EXEC clistname" in the command line). For further information, please refer to the relevant IBM manuals.

2.3 Installation of the openFT-CR delivery unit

If openFT is also to be able to transfer job data (file contents) in encrypted form in file transfer requests, the openFT-CR delivery unit must be installed.

The openFT-CR delivery unit is supplied on a separate product tape (and as a separate order) free-of-charge.

The delivery unit openFT-CR is installed with SMP/E as a supplement to the function SYSMOD for openFT described in section “Reading in the openFT product tape” on page 36. So, in order to be able to install the delivery unit openFT-CR, openFT must be available in an SMP/E environment.

For installation, the product data is also offered on CD as well as product tape. You will find further information and installation instructions on the CD.

2.3.1 Contents of the product tape

The product tape, which contains the openFT-CR data sets, is created as an "SL tape" using the IBM utility IEBCOPY. It contains the following files in the order in which they are listed below:

OPENFT.OFT110A.SMPMCS
MCS statements for SMP/E

OPENFT.OFT110A.F1
PO data set containing the following CLIST's:
OFC110A JCLIN statements for transferring the other product modules from the tape with SMP/E
OPFTCR#1 creates the system and backup files for openFT-CR
OPFTCR#2 extends the SMP/E environment for openFT by the entries required for openFT-CR
OPFTCR#3 executes the SMP/E statement RECEIVE
OPFTCR#4 executes the SMP/E statement APPLY
OPFTCR#5 executes the SMP/E statement ACCEPT

OPENFT.OFT110A.F2
PO data set containing the following members:
FGMD#CR contains the Release Notice in printable form in German
FGME#CR contains the Release Notice in printable form in English

OPENFT.OFT110A.F3
Load module library containing the following member:
OPENFTCR Load module for openFT-CR

2.3.2 How to proceed with the installation

Proceed as follows to install openFT-CR:

1. Copy the second file (OPENFT.OFC100A.F1) from the product tape to a PO data set using IEBCOPY (a sample job on page 39, where the name of the file being copied needs to be modified).

2. Adjust the CLISTs in members OPFTCR#1 to OPFTCR#5 of the copied PO data set to the computer center. You can, for example use the T_BASE variable to assign the installation location for openFT-CR.

3. Execute the modified CLISTs:

   OPFTCR#1 creates the system and backup files for openFT -CR
   OPFTCR#2 extends the SMP/E environment for openFT by the entries needed for openFT-CR
   OPFTCR#3 executes the SMP/E statement RECEIVE
   OPFTCR#4 executes the SMP/E statement APPLY
   OPFTCR#5 executes the SMP/E statement ACCEPT

4. The load module OPENFTCR contained in the library OPENFT.OFC110A.F3 must be copied to a member with the same name in the load module library that also contains the load module of the openFT program, or to a concatenated library.
   To load this module, openFT calls the system macro LOAD, which searches for a member with the name OPENFTCR in the conventional library hierarchy STEPLIB, TASKLIB, JOBLIB ... . If openFT is to run with "APF authority", the library that contains the OPENFTCR load module (i.e. the one to which the OPENFTCR member was copied from the product tape) must also be APF-authorized (see section “openFT privileges” on page 31).

   The OPENFTCR module can be added or removed when the local openFT instance is deactivated, i.e.:

   – openFT must be stopped using the FTSTOP command and

   – the started openFT job must be terminated using the FTTERM command or the started openFT task must be terminated.

   If the local openFT is then restarted, i.e. the openFT job is loaded with the FJINIT command or openFT is restarted as a "started task" and openFT is then activated with FTSTART, openFT searches for the member in the library hierarchy given above.

   If the load module OPENFTCR is not contained in the openFT load library (or a concatenated library), the function "encoded transfer of job data" cannot be used. Depending on your system environment, the following system message is output to the job protocol of openFT after openFT is activated (command FTSTART):

   ```
   CSV003I REQUESTED MODULE OPENFTCR NOT FOUND
   ```

2.4 Installation of the openFT-AC delivery unit

If openFT is to be used with the FTAC functionality (see section “Administrating and controlling FTAC functions” on page 126), the openFT-AC delivery unit must be installed.

The openFT-AC delivery unit is supplied on a separate product tape (and must be ordered separately).

The openFT-AC delivery unit is installed with SMP/E as an supplement to the function SYSMOD for openFT described in section “Reading in the openFT product tape” on page 36. openFT must be available in an SMP/E environment if the openFT-AC delivery unit is to be installed.

> **i** For installation, the product data is also offered on CD as well as product tape. You will find further information and installation instructions on the CD.

2.4.1 Contents of the product tape

The product tape, which contains the openFT-AC data sets, is created as an "SL tape" using the IBM utility IEBCOPY. It contains the following files in the order in which they are listed below:

OPENFT.OFA110A.SMPMCS
MCS statements for SMP/E

OPENFT.OFA110A.F1
PO data set containing the following CLIST's:

OFA110A	JCLIN statements for transferring the other product modules from the tape with SMP/E
OPFTAC#1	creates the system and backup files for openFT-AC
OPFTAC#2	extends the SMP/E environment for openFT by the entries needed for openFT-AC
OPFTAC#3	executes the SMP/E statement RECEIVE
OPFTAC#4	executes the SMP/E statement APPLY
OPFTAC#5	executes the SMP/E statement ACCEPT

OPENFT.OFA110A.F2
PO data set containing the following members:

FGMD#AC	contains the Release Notice in printable form in German
FGME#AC	contains the Release Notice in printable form in English

OPENFT.OFA110A.F3
Load module library containing the following member:

OPENFTAC	Load module for openFT-AC

2.4.2 How to proceed with the installation

Proceed as follows to install openFT-AC:

1. Copy the second file (OPENFT.OFA110A.F1) from the product tape to a PO data set using IEBCOPY (a sample job on page 39, where the name of the file being copied needs to be modified).
2. Adjust the CLISTs in members OPFTAC#1 to OPFTAC#5 of the copied PO data set to the computer center. You can, for example use the T_BASE variable to assign the installation location for openFT-AC.
3. Execute the modified CLISTs:

   OPFTAC#1 creates the system and backup files for openFT-AC
   OPFTAC#2 extends the SMP/E environment for openFT by the entries required for openFT-AC
   OPFTAC#3 executes the SMP/E statement RECEIVE
   OPFTAC#4 executes the SMP/E statement APPLY
   OPFTAC#5 executes the SMP/E statement ACCEPT

   The FTAC command entries created at the time of openFT installation can be used following installation of openFT-AC.

2.5 Installing the openFT-FTP delivery unit

If you wish to use openFT with the FTP functionality, you must install the delivery unit openFT-FTP.

The delivery unit openFT-FTP is supplied on a separate product tape (to be ordered separately).

The openFT-FTP delivery unit is installed using SMP/E as a supplement the "function SYSMOD" for openFT described in section “Reading in the openFT product tape” on page 36. This means that openFT must be available in an SMP/E environment to be able to install openFT-FTP.

ℹ For installation, the product data is also offered on CD as well as product tape. You will find further information and installation instructions on the CD.

2.5.1 Contents of the product tape

The product tape on which the data sets for openFT-FTP V11.0A are supplied is created as an "SL tape" using the IBM utility IEBCOPY and contains the following files in the order listed here:

OPENFT.OFP110A.SMPMCS
MCS statements for SMP/E

OPENFT.OFP110A.F1
PO data set containing the following CLISTs:

OFP110A	JCLIN statements for transferring the other product modules from the tape with SMP/E
OPFTP#1	creates the system and backup files for openFT-FTP
OPFTP#2	extends the SMP/E environment for openFT-FTP to include the entries required for openFT-FTP
OPFTP#3	executes the SMP/E statement RECEIVE
OPFTP#4	executes the SMP/E statement APPLY
OPFTP#5	executes the SMP/E statement ACCEPT

OPENFT.OFP110A.F2
PO data set containing the following members:

FGMD#FTP	Release Notice in German
FGME#FTP	Release Notice in English

OPENFT.OFP110A.F3
load module library containing the following members:

OPENFTP	load module for openFT-FTP

2.5.2 How to proceed with the installation

Proceed as follows to install openFT-FTP:

1. Copy the second file (OPENFT.OFP110A.F1) from the product tape to a PO data set using IEBCOPY.
2. Adjust the CLISTs in members OPFTP#1 to OPFTP#5 of the copied PO data set to the computer center. You can, for example, use the T_BASE parameter to assign the installation location for openFT. By default openFT is installed at the location to which the XMIT files were copied for installation purposes.
3. Execute the modified CLISTs:

   OPFTP#1 creates the system and backup files for openFT-FTP
   OPFTP#2 extends the SMP/E environment for openFT to include the entries required for openFT-FTP
   OPFTP#3 Executes the SMP/E statement RECEIVE
   OPFTP#4 Executes the SMP/E statement APPLY
   OPFTP#5 Executes the SMP/E statement ACCEPT

   The OPENFTP load module contained in the library OPENFT.OFP110A.F3 is installed under the same name as a member of the same library that contains the load modules for openFT.

i If you want to use the openSSL functionality in addition to the FTP functionality offered by the installed unit openFT-FTP, the following line in the batch job or the "started task"

```
//OPENFT EXEC PGM=OPENFT,TIME=1440,
```

must be changed to

```
//OPENFT EXEC PGM=OPENFTSL,TIME=1440,
```

See also “Example of the FJBATCH member” on page 86.

2.6 Startup

A number of steps are required for initial startup of openFT:

- Set the necessary installation parameters with FJGEN, see below.
- Set up the FT parameter library, see page 49.
- Make the OPFT subsystem available, see page 85.
- Specify whether openFT is to run as a job or a started task, see page 85.
- Load and start the openFT load module (see page 89) if openFT is not to be started as a started task.

You can then start, stop and terminate openFT, see page 89.

2.6.1 Setting openFT installation parameters with FJGEN

You use the TSO procedure FJGEN from the library OPENFT.CLIST to set up a new openFT instance or, subsequently, to modify the parameter settings of existing instances. In your openFT system, instances are identified via their instance names which you specify in INSTANCE NAME during FJGEN processing. FJGEN expects the load modules of openFT to be specified in the libraries OPENFT.LOAD and OPENFT.NCLOAD.

FJGEN runs a dialog to query the parameters to be defined during the first installation step. In particular, the qualifier under which the openFT administrative files and the instance-specific FT procedure library are stored is defined here. (OPENFT QUALIFIER).

The FJGEN command is required even if the installation parameters have been set using a file (see section “Setting up the FT parameter library” on page 49). If openFT is to run as a started task, the FT administrator must create the start procedure himself and specify the corresponding values for the installation parameters (see section “openFT as a job or started task” on page 85). Here also, however, the FJGEN command is required to create further command procedures and create instance-specific files.

The FJGEN command can also be used to change the installation parameters of an openFT instance. The named procedures are then regenerated in the instance-specific procedure library. The changes become effective the next time the installation-specific batch job is started with FJINIT. (If openFT is running as a started task, the FT administrator is responsible for making the corresponding changes in the start procedure, see section “openFT as a job or started task” on page 85). When an instance-specific procedure library is concatenated, you can obtain information about the current values of the installation parameters using the "FJGENPAR" command.

A detailed description of the FJGEN command and examples are given on page 182.

2.6.2 Setting up the FT parameter library

You can use the FT parameter library to tailor openFT to the specific requirements of your installation. You can store the following information in the members of this library:

- openFT Installation parameters: Some installation parameters are also specified in the FJGEN command, see page 182 and page 48; installation parameters entered in the FT parameter library take precedence over those specified in FJGEN.)
- Definitions of the users (user IDs) who possess FT or FTAC administrator rights
- Job cards for printing the result list and preprocessing, postprocessing and follow-up processing.
- Address information from an older openFT version that is still to be used in openFT V11.
- Details on the use of file-specific character sets (see section “Structure of the member FNAMECTB” on page 80 and section “Administering code tables” on page 103)
- Installation parameters required when openFT-AC is used
- Specifications for creating diagnostics information.

Format and name of the FT parameter library

The information from the FT parameter library is stored as text in members of a PO or PDSE dataset. By default, the library must be created under the following name:

<openft qualifier>.<inst>. PARM

The OPENFT QUALIFIER (<openft qualifier>) and the INSTANCE NAME(<inst>) are defined using FJGEN (see page 182).

The following also applies:

- It is advisable to create this data set with the record format F or FB and a record length of 80. openFT fills records with a length of less than 80 in order to make them 80 characters long. Records whose length exceeds 80 characters are truncated after the 80th character. This can lead to errors if invalid job cards are created as a result, particularly for the members TSOJOB, TSOVVJOB, TSONVJOB, TSOVFJOB, JCLJOB and PRTJOB (see below).
- The members of the FT parameter library may **not** contain line numbering. Please observe this rule when creating or editing the text contained in the members. (If you use the PDF editor, e.g. via the menu interface for the FT administrator, you must therefore set NUMBER OFF in your EDIT profile)

Elements of the FT parameter library

PARM:

Installation parameters for openFT. The structure of this member is described on page 52.

FTADM:

List of users with FT administration authorization. The structure of this member is described on page 62.

FTACADM:

List of users with FTAC administration authorization. The structure of this member is described on page 62.

PRTJOB:

Job cards for printing the result list. The structure of this member is described on page 63.

JCLJOB:

Job cards for the follow-up job which is created by openFT if the follow-up processing consists of one or more JCL statements. The structure of this member is described on page 63.

TSOJOB:

Job cards for the follow-up job which is created by openFT if the follow-up processing consists of one or more TSO commands. The structure of this member is described on page 63.

TSOVVJOB:

Job cards for the preprocessing job generated by openFT if one or more TSO commands have been requested as preprocessing commands in an FT request. The structure of this member is described on page 63.

TSOVFJOB:

Job cards for the preprocessing job generated by openFT if the "ftexec" command is issued for the z/OS system. The structure of this member is described on page 63.

TSONVJOB:

Job cards for the postprocessing job generated by openFT if one or more TSO commands have been requested as postprocessing commands in an FT request. The structure of this member is described on page 63.

SUCCMSG:

Text of the asynchronous message which openFT issues as a result of successful file transfer to one or several consoles. This member is evaluated only if a valid specification has been made for the SUCC_MSG keyword in the FTMSPPAR member. The structure of this member is described on page 74.

FAILMSG:

Text of the asynchronous message which openFT issues as a result of unsuccessful file transfer to one or several consoles. This member is evaluated only if a valid specification has been made for the FAIL_MSG keyword in the FTMSPPAR member The structure of this member is described on page 74.

TNSTCPIP:

Address information from an older version of openFT that is still to be used in openFT V11. This element is only supported for reasons of compatibility.

FNAMECTB:

Information on of which file-specific code character sets (see section “Administering code tables” on page 103) openFT is to use and which files are to be coded with the relevant character set. The structure of the member is described on page 80.

FTACPAR:

Installation parameter needed when openFT-AC is used. The structure of the member is described on page 84.

DIAGPAR:

Specifications for creating diagnostics information. The structure of the member is described on page 141.

The members PARM, PRTJOB, JCLJOB, TSOJOB, TSOVVJOB, TSONVJOB, TSOVFJOB, SUCCMSG, FAILMSG, FTADM, FTACADM, FNAMECTB, FTACPAR and DIAGPAR are read and evaluated when the openFT load module is started, i.e. when the openFT batch job is started (with FJINIT) or when the started task commences. Errors in accessing a member are not reported. openFT acts as though the member concerned were not present. In this case the appropriate default values are used, if any are available.

The TNSTCPIP element is read in and evaluated when the local openFT instance is activated (FTSTART). Errors on access to the TNSTCPIP member are notified with the error message FTR4040 (see page 432). Errors on access to the CLASSDEF and CLASSATT members are not notified.

Any updates to the TNSTCPIP, FTADM and FTACADM members can be read in during system operation using the FTUPDPAR command (see page 365).

2.6.2.1 Structure of the PARM member

You can specify installation parameters for openFT in this element.

The description of the FJGEN command (page 182) shows the structure of the batch job which is required for openFT to run as a background task (FJBATCH member of the FT procedure library). This job remains unchanged even if you specify installation parameters in the PARM member of the FT parameter library. In particular, the string containing the start parameters in the batch job remains unchanged. If, however, you also specify a corresponding parameter in the PARM member, openFT uses this value; the value from the start parameters is not used in this case.

Each line of the PARM member can contain exactly one parameter in the form "keyword=value". No blanks may be inserted between "keyword", "=" and "value". Below is a list of the keywords which may be used.

Keywords:

DESTVOL=
: Definition of the volume for local receive files which do not exist and for files used to store the result lists (LISTING=*LISTFILE). Exactly 6 characters. See also the description of the DESTUNIT parameter.

DESTUNIT=
: Definition of the unit/groupname for local receive files which do not exist and for files used to store result lists (LISTING=*LISTFILE). Maximum 6 characters.

If you specify a value for only one of the parameters DESTVOL and DESTUNIT, openFT assigns blanks to the other. If you do not specify a value for either of the parameters DESTVOL and DESTUNIT, receive files which do not exist and files for result lists are created on the default volume (system-specific).

UNLOADVOL=
: Definition of the volume for temporary PS data sets with the suffix ".U" to which the entire PO/PDSE data sets are transferred (in "unloaded" format) prior to file transfer. Exactly 6 characters. See also the description of the UNLOADUNIT parameter.

UNLOADUNIT=
: Definition of the unit/groupname for temporary PS data sets with the suffix ".U" to which entire PO/PDSE data sets are transferred (in "unloaded" format) prior to file transfer. Maximum 6 characters.

If you specify a value for only one of the parameters UNLOADVOL and UNLOADUNIT, openFT assigns blanks to the other. If you do not specify a value for either of the parameters UNLOADVOL and UNLOADUNIT, the temporary PS data sets are created on the default volume (system-specific).

In order to transfer entire PO/PDSE data sets, openFT must be APF-authorized (see section “openFT privileges” on page 31).

PODIR=

Number of directory blocks which are to be reserved by openFT when creating a PO data set. Maximum value: 32767 (default: 20).

DSTYPEDEF=

Default value for file organization if the receive file of a file transfer request is to be created as a sequential file whose file organization is not precisely defined by the structure of the send file (if homogeneous systems are used as of V10 partners) or the file name (see User Guide „openFT for z/OS and z/OS - Managed File Transfer in the Open World").
Possible Values:

PS	A "physical sequential data set" (PS data set) is created.
VSAM	A VSAM file of type "entry sequenced" is created.

If no value or an invalid value is specified for the DSTYPEDEF parameter, the default value PS applies.

LIBTYPEDEF=

Default value for the file organization if a data set (library) divided into members is to be created for a file transfer request in the receive system and the file organization is not precisely defined by the structure of the send file (if a complete PO or PDSE data set is transferred as of V10 partners) or the file name (see User Guide „openFT for z/OS - Managed File Transfer in the Open World").
This value is important

– when a member is the receive file, the associated data set does not yet exist and the type (PO/PDSE) has not been specified (prefix :L: or no prefix),
– when a complete PO or PDSE data set is the receive file, the type (PO/PDSE) has not be specified precisely (prefix :L:).

Possible values:

NONE openFT does not specify the file organization, the system defaults therefore apply:
  – if the IBM software product DFSMS (Data Facility System-Managed Storage) is not installed, PO is the default
  – if DFSMS is active: definition of the default by the ACS routine or SYS1.PARMLIB (please ask your z/OS system administrator)

PO A "partitioned organized data set" (PO data set) is created.

PDSE An attempt is made to create a "partitioned organized data set extended" (PDSE data set). This is only possible if the IBM software product DFSMS is installed and the parameters PDSESTORC, PDSEMGMTC and PDSEDATAC (see below) have been set correctly.

If no value or an incorrect value is specified for LIBTYPEDEF, the default value NONE applies.

PDSESTORC=

SMS storage class for PDSE data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS storage class.
The value is only used if there are no settings concerning the SMS storage class on your system (please ask your z/OS system administrator).

Please observe the description of the PDSEDATAC parameter.

PDSEMGMTC=

SMS management class for PDSE data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS management class.
This value is only used if there are no settings concerning the SMS management class on your system (please ask your z/OS system administrator).

Please observe the description of the PDSEDATAC parameter.

PDSEDATAC=

SMS data class for PDSE data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS data class.
This value is only used if there are no settings concerning the SMS data class on your system (please ask your z/OS system administrator).

The parameters PDSESTORC, PDSEMGMTC and PDSEDATAC only become effective if a PDSE data set is to be newly generated as a receive file (either because a complete PDSE file set was specified as the receive file or because a PDSE member is a receive file, but the relevant PDSE data set does not exist yet).

The parameters all only become effective if your system does not contain any specifications for the relevant SMS class.

As a rule, you should not specify any of these parameters; the settings made in your system will then apply. Exception: if no settings are made in your system for any of the SMS classes, you must specify at least one of the parameters PDSESTORC, PDSEMGMTC and PDSEDATAC, otherwise openFT will not be able to generate a PDSE data set.

An invalid specification for one of the parameters, which will become effective (because there are no default settings for the relevant SMS class in your system) will cause those transfer requests to fail for which a new PDSE data set has to be generated on the receive system. Further details are contained in the manual "openFT for z/OS - Managed File Transfer in the Open World").

POSTORC=

SMS storage class for PO data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS storage class.
This value is only used if there are no settings concerning the SMS storage class on your system (please ask your z/OS system administrator).

Please observe the description of the PODATAC parameter.

POMGMTC=

SMS management class for PO data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS management class.
This value is only used if there are no settings concerning the SMS management class on your system (please ask your z/OS system administrator).

Please observe the description of the PODATAC parameter.

PODATAC=

SMS data class for PO data sets (refer to the literature on the IBM software product DFSMS for further details).
Maximum 8 characters; valid name of an SMS data class.
This value is only used if there are no settings concerning the SMS data class on your system (please ask your z/OS system administrator).

If PO data sets are to be created as SMS-managed data sets (prerequisite for this is that the IBM software product DFSMS is installed), the same in essence applies for parameters POSTORC, POMGMTC and PODATAC as for parameters PDSESTORC, PDSEMGMTC and PDSEDATAC (see description of parameter PDSEDATAC on page 54).

JOB_MSGCLASS=

Message class of the follow-up processing job. (This is the default value for the JOB statement parameter MSGCLASS= if the members TSOJOB or JCLFOB contain no relevant entry.) Exactly 1 character (default value: A).

LST_MSGCLASS=

Message class of the job for printing the result list. (This is the default value for the JOB statement parameter MSGCLASS= if the member PRTJOB contains no relevant entry.) Exactly 1 character (default value: A).

NABVOLUME=

The volume on which the request file, the partner list, the operating parameter file, the logging file and the FTAC file are to be located (see section “Internal openFT data sets” on page 424).
Exactly 6 characters.

If the corresponding files are SMS managed, the specifications for Volume and Unit may have no effect under certain circumstances. If the files are not SMS managed, an "SMS managed volume" must not be specified here.

See also the description of the NABUNIT parameter.

NABUNIT=

Definition of the unit/groupname of the volume on which request file, the partner list, the operating parameter file, the logging file and the FTAC file is to be set up. Maximum 6 characters.

If you specify a value for only one of the parameters NABVOLUME and NABUNIT, openFT assigns blanks to the other.

If you do not specify a value for either of the parameters NABVOLUME and NABUNIT, the values of DMP_VOLUME and DMP_UNIT (see below) are assumed. Either or both of these values may in turn have been taken from the values specified for VOLUME/UNIT in the FJGEN command.

LOGFILE_2ND_Q=

The second level qualifier for creating the names of the components of the logging file (see section “Internal openFT data sets” on page 424).
Up to 18 characters (default: <inst>.SYSLOG, where <inst> is the name of the openFT instance). For the sake of clarity, the name should always start with the instance name followed by a period.

This name must be specified in partially qualified form, i.e. with no "first level qualifier" or single quotes. openFT prefixes this name with the OPENFT QUALIFIER specified in FJGEN.

LOGFILE_SIZE_RC=

Initial size of the logging file (number of logging records).
Maximum value: 16777215 (default: 10000).

openFT uses this value as the primary allocation when creating the VSAM cluster which is part of the logging file. For the secondary allocation, the value is halved.

The specified maximum value is the program-technical limit. When choosing a value for the initial size of the logging file, the actually available storage space needs to be taken into consideration. Note that the logging file is created on the same data volume as the request file, the partner list and (if FTAC is installed) the FTAC file (see section “Internal openFT data sets” on page 424).

DMP_VOLUME=

Definition of the volume on which openFT creates the dump and trace files. Exactly 6 characters (default: value specified for VOLUME in the FJGEN command; see the description of the FJGEN command, page 182). You can also use "DMP_VOLUME= " to specify that the value specified for VOLUME in the FJGEN command is not used when the dump and trace files are created. See also the description of the DMP_UNIT parameter.

DMP_UNIT=

Definition of the unit of the volume on which openFT creates the dump and trace files. Maximum 6 characters (default: value specified for UNIT in the FJGEN command (page 182).

You can also use "DMP_UNIT= " to specify that the value specified for UNIT in the FJGEN command is not used when the dump and trace files are created.

If, after evaluation of the specifications for FJGEN (VOLUME/UNIT) and the specifications made here, there is no value for DMP_VOLUME or for DMP_UNIT, openFT uses the UNIT name DASD. This UNIT name must then be defined in the system.

Examples

VOLUME/UNIT (FJGEN)	**FTMSPPAR**	**Result**

VSN123/SYSDA	DMP_VOLUME=VSN456	DMP_VOLUME = VSN456
		DMP_UNIT = SYSDA

VSN123	DMP_UNIT=SYSDA	DMP_VOLUME = VSN123
		DMP_UNIT = SYSDA

/	DMP_VOLUME=VSN456
	DMP_UNIT=SYSDA
		DMP_VOLUME = VSN456
		DMP_UNIT = SYSDA

VSN123/SYSDA	DMP_VOLUME=	DMP_VOLUME = no value
		DMP_UNIT = SYSDA

VSN123	DMP_VOLUME=	DMP_VOLUME = no value
		DMP_UNIT = DASD (!)

/	no specifications	DMP_VOLUME = no value
		DMP_UNIT = DASD (!)

ROUTCDE=

Routing code of one console to which the openFT asynchronous messages are to be output. (Note openFT uses the WTO macro to output these messages to the console.) Valid values: 1 through 128. Invalid values are ignored and no message is issued.

SMF_RECORD_TYPE=

Type of the accounting record written by openFT to the SMF file. Valid values:

128 through 255
For each transfer request accepted, an accounting record of the specified type is written to the SMF file, provided that SMF is active. The structure of the accounting records is described in the appendix.

0
No accounting records are written to the SMF file. (Default; values outside the valid range are interpreted as 0.)

SMF_ADM_AREA=

Installation-specific text written by openFT to the FT administrator area of the accounting records (see the description of the accounting record structure in the section “Accounting records” on page 408). This text may be up to 40 characters long. Default: blanks.

In order to enter SMF accounting information, openFT must be APF-authorized (see section “openFT privileges” on page 31).

SUCC_MSG=

Specifies when an asynchronous message indicating successful file transfer is to be issued. Valid values:

IN
An asynchronous message indicating successful file transfer is output only for transfer requests which were submitted in a remote system.

OUT
An asynchronous message indicating successful file transfer is output only for transfer requests which were submitted in the local system.

BOTH
An asynchronous message is output for all transfer requests following successful file transfer.

Invalid values are ignored and no error message is issued. In this case, no asynchronous message is output following successful file transfer (default value).

The destination for output of the asynchronous message after successful file transfer is controlled by the keyword ENDMSG_ROUTCDE (see below).

The message text for the asynchronous message following successful file transfer can be defined in the SUCCMSG member of the FT parameter library; otherwise, openFT uses a standard text (see page 74).

In order to output asynchronous messages following termination of a request, openFT must be APF-authorized (see section “openFT privileges” on page 31).

FAIL_MSG=

Specifies when an asynchronous message indicating unsuccessful file transfer is to be output. Valid values:

IN	An asynchronous message indicating unsuccessful file transfer is output only for transfer requests which were submitted in a remote system.
OUT	An asynchronous message indicating unsuccessful file transfer is output only for transfer requests which were submitted in the local system.
BOTH	An asynchronous message is output for all transfer requests following unsuccessful file transfer.

Invalid values are ignored and no error message is issued. In this case, no asynchronous message is output following unsuccessful file transfer (default value).

The destination for output of the asynchronous message after successful file transfer is controlled by the keyword ENDMSG_ROUTCDE (see below).

The message text for the asynchronous message following unsuccessful file transfer can be defined in the FAILMSG member of the FT parameter library; otherwise, openFT uses a standard text (see page 74).

In order to output asynchronous messages following termination of a request, openFT must be APF-authorized (see section “openFT privileges” on page 31).

ENDMSG_TO_TSO=

Switch for controlling the output of asynchronous messages to a TSO terminal at the end of a job. The output is made to the terminal of the TSO user whose user ID was specified in the TRANSFER-ADMISSION. The messages are only output for jobs issued locally; You can find them in the appendix as of page 438.
Possible values:

YES	(Default): The asynchronous messages are output.
NO	(Or invalid value): The asynchronous messages are not output.

The asynchronous messages output to a TSO terminal at the end of a job also appear in the openFT job protocol (see page 414).

ENDMSG_ROUTCDE=

Routing code of one console to which the asynchronous messages are to be output at the end of a job. The cases in which an asynchronous message is output at the end of a job is controlled by the keywords SUCC_MSG and FAIL_MSG (see above).
Possible values: 1 to 128. Invalid values are ignored and no message output.

If the keyword is missing or assigned no (or an invalid) value, no output is made to the console.

The asynchronous messages output to a console at the end of a job are assigned a key (FJM2100 for the message following successful file transfer); this makes it possible to process these messages with NetView. The messages also appear in the job protocol openFT in this form (see page 414).

TCP_USERID=

Name of the TCP/IP address space. If the name of the TCP/IP address space is not TCPIP (default), you must specify it here. Message FTR4055 can indicate that the name of the TCP/IP address space has not been specified correctly. Ask your z/OS system administrator.

Up to 8 characters (default: TCPIP).

MSG_CRYPT=

Optionally, it is possible to encrypt the messages from the openFT dialog tasks for the purposes of internal communications with the openFT subsystem. Commands are always encrypted. The mechanism employed is the same as for the encryption of the request description data.
Valid values:

Y	Messages are encrypted.
N	Messages are not encrypted (default).

CMD_TRANS=

You use this switch to define the transport protocol to be used to connect the dialog tasks to the openFT subsystem. If openFT implicitly recreates the parameter library then CMD_TRANS=TCP is preset.
Valid values:

VTAM	Communication is performed via VTAM.
TCP	Communication is performed via TCP (default)

OPENFT_SVC=

The openFT subsystem administers all the running instances and encrypts or decrypts all the commands, messages (optional, see the MSG-CRYPT parameter) and connection data. The portal to the subsystem is implemented via SVC 109 with "extended code 211". This ESR SVC code is defined using OPENFT-SVC. If "extended Code 211" is already used for a different purpose in your system, you can use the LINK procedure LINKIGX from the SAMPLES library in order to utilize your extended code. For reasons of security, OPENFT-SVC should be set to a valid value.

USER_INACT_TIME

This specification defines a maximum idle time (in minutes) before a connection between the user TSO interface and the openFT subsystem will be terminated for security reasons.
Valid values:

0 ..30	Time specification for the maximum idle time in minutes. (Default: 5 minutes)

PSSTORC=

SMS storage class for PS datasets. For further details, see the documentation on the IBM software product DFSMS.

Up to 8 characters; valid name of an SMS storage class.
This value only has any effect of there is no default specification for the SMS storage class in your system (ask your z/OS system administrator).
Refer also to the description of the PSDATAC parameter.

PSDATAC=

SMS data class for PS datasets. For further details, see the documentation on the IBM software product DFSMS.

Up to 8 characters; valid name of an SMS data class.
This value only has any effect of there is no default specification for the SMS data class in your system (ask your z/OS system administrator).

i The PSSTORC and PSDATAC parameters only take effect if a new PS dataset is to be created as the receive file. These parameters are also valid for creating trace files.

DEFFSIZE=

Size of a secondary allocation for the receive file if the size of the send file is unknown. DEFFSIZE is specified in bytes. In this case, the primary allocation is approximately one tenth of this value. If this specification is omitted, DEFFSIZE=2621440 is taken. DEFFSIZE also influences the primary and secondary allocations if the size of the send file is specified. For further details, refer to the section "File types - z/OS files" in the User Guide.

MAXALLOC=

Maximum size of file allocations (both primary and secondary). MAXALLOC is specified in megabytes. The default value is 1024, and the (theoretical) maximum value is 32767.

Example of the member PARM

```
DESTVOL=TSO000
DESTUNIT=SYSDA
UNLOADVOL=TSO000
UNLOADUNIT=SYSDA
DSTYPEDEF=PS
LIBTYPEDEF=PO
JOB_MSGCLASS=X
LST_MSGCLASS=X
SUCC_MSG=BOTH
FAIL_MSG=BOTH
ENDMSG_TO_TSO=YES
TCP_MYPORT=1100
LOGFILE_2ND_Q=OPENFTLG
```

2.6.2.2 Structure of the members FTADM and FTACADM

The members FTADM and FTACADM contain all the users (user IDs) who possess FT or FTAC administrator authorization. Each entry must start on a new line in column 1. User ID groups that differ only in the associated suffix and which all possess the same authorizations can be combined using wildcards "*". For example, the user IDs XORG001, XORG002 and XORG003 can be represented by a single entry XORG*.

If you enter administration commands at the console or use Netview then a pseudo-entry *Console* must be set up in these members.

In FTACADM, it is possible to assign FTAC administrators what FTAC considers to be "system administrator rights". The restrictions applying to the setup and import of admission profiles for external user IDs do not apply to these administrators. To assign this privilege, enter SU after the user ID in column 10/11.

⚠ **WARNING!**
FTAC administrators with the "SU privilege" can set up appropriate admission profiles allowing them to access the files belonging to any user ID and, in this way, circumvent any protection policies that may be in place! For this reason, it is necessary to treat write access rights to the FT parameter library with considerable care.

If the PARM library does not exist at the time the FJGEN command is called, openFT creates the members FTADM and FTACADM during FJGEN execution. These are then assigned the OPENFT USER ID and the pseudo-entry "Console". The members may be modified (e.g. other user IDs may be entered in them). Changes take effect the next time openFT is loaded or when the FTUPDPAR command is issued. FTADM and FTACADM may each contain up to 100 entries.

2.6.2.3 Structure of the members PRTJOB, JCLJOB, TSOJOB, TSOVVJOB, TSOVFJOB and TSONVJOB

Each of these members consist of prototype statements which openFT uses if it is creating an appropriate job internally. openFT does not check the syntax of these prototype statements. Sample members are supplied in the library SAMPLES that is delivered with openFT.

Each record contains exactly one job card (or continuation card). A maximum of 32767 records are evaluated for each member.

The following variables can be used in these prototype statements. openFT replaces these symbolic parameters with the current values:

JOBP

Job name prefix, identical to USID if the user ID does not exceed 7 characters in length. Otherwise, the last character is removed.

USID

User ID from TRANSFER-ADMISSION (for TSOVVJOB, TSOVFJOB and TSONVJOB) or from PROCESSING-ADMISSION (for JCLJOB and TSOJOB).
Maximum 8 characters, in accordance with IBM conventions.

ACCN

"accounting information" from TRANSFER-ADMISSION (PRTJOB, TSOVVJOB, TSOVFJOB and TSONVJOB) or from PROCESSING-ADMISSION (for JCLJOB and TSOJOB).
Maximum 40 characters, in accordance with IBM conventions.

PASS

Password from TRANSFER-ADMISSION (PRTJOB, TSOVVJOB, TSOVFJOB and TSONVJOB) or from PROCESSING-ADMISSION (for JCLJOB and TSOJOB).
Maximum 8 characters, in accordance with IBM conventions.

OWID

Owner of the FT request, i.e. the user ID under which the transfer job was created.
Maximum 8 characters according to IBM conventions.

This variable is replaced only in the system where the transfer request was issued. It is eliminated in the remote system.

PGRN

„programmer's name" as specified a subcommand in the relevant command string for follow-up processing with the keyword PGRN= (see User Guide "openFT for z/OS - Managed File Transfer in the Open World"). The PRTJOB member also assigned the value from the relevant command string for follow-up processing, i.e. from the command

string in the SUCCESS-PROCESSING on successful processing and from the command string in the FAILURE-PROCESSING if processing failed.
Maximum 20 characters in accordance with IBM conventions.

If there is no value for "programmer's name" and, after replacement of the PGRN variables inside a JOB Statement, it is established that the corresponding card has no other data except for the JCL identifier „//" and comma (separator for parameters), this card is ignored, i.e. there is no execution. This is in keeping with the recommendation in the IBM literature (JCL Reference) not to mark this missing parameter with a comma.

TRID

FT transfer ID.
Maximum 10 characters (value range 1..2147483639) in accordance with openFT conventions.

RLFN

Name of the file in which the result list is stored (PRTJOB). Maximum of 56 characters, in accordance with IBM conventions (the maximum length is obtained from the specified structure of this file name; refer to the "openFT for z/OS - Managed File Transfer in the Open World" for further details).
This variable can be used in the member PRTJOB; in other members it is removed.

RLFP

Temporary file to which the preprocessing operation outputs your data (TSOVVJOB and TSOVFJOB). Maximum 38 characters in accordance with IBM conventions. This variable can only be used in the members TSOVVJOB and TSOVFJOB. In other members, it is removed.

RLFF

Temporary file to which the preprocessing operation of an ftexec command outputs its error messages. Maximum 38 characters in accordance with IBM conventions. This variable can only be used in the member
TSOVFJOB. In other members, it is removed.

RLFT

Temporary file to which the preprocessing operation of an ftexec command issued in a Unix or Window partner system outputs its TSO messages. Maximum 38 characters in accordance with IBM conventions. This variable can only be used in the member TSOVFJOB. In other members, it is removed.

CONN

Name of the file containing the key for the connection to the FT subsystem. Maximum 36 characters in accordance with IBM conventions. By default, this file is created under <openft qualifier>.<inst>.CONN.This variable can only be used in the members TSOJOB, TSONVJOB, TSOVVJOB and TSOVFJOB. In other members, it is removed.

NCLO

Name of the file that contains the openFT commands. Maximum 36 characters in accordance with IBM conventions. By default, this file is created under OPENFT.NCLOAD. This variable can be used in the members TSOJOB, TSONVJOB, TSOVVJOB and TSOVFJOB. It is eliminated from other members.

The names of these variables should have as many trailing "#" fill characters as are necessary for a field to be set to its maximum length (including the "&" character, e.g. &TRID#####). When replacing the variables by the current values, openFT does not exceed the field length predefined by the name of the symbolic parameter including the trailing "#" fill characters; if necessary the current values are truncated. On the other hand, where the current values are shorter than this field length, openFT removes superfluous fill characters.

Note

If a follow-up processing job of the type ALLOC DSNAME (...) was specified, openFT also replaces the variables in this job before passing it to the Internal Reader.

When creating sample instructions, the requirements of the relevant z/OS installation for executable jobs must be observed. As a rule, the JOB statement requires a valid user ID, valid accounting information and a valid user password. These values can be taken from the following sources:

– They are specified by the user in the NCOPY command (for PRTJOB, TSOVVJOB, TSOVFJOB and TSONVJOB in the TRANSFER-ADMISSION, for JCLJOB and TSOJOB in the PROCESSING-ADMISSION).

– If FTAC is used, the specifications for PROCESSING-ADMISSION can also be defined within an admission profile. The admission to perform preprocessing and postprocessing is defined via the specifications for the TRANSFER-ADMISSION.

– They are contained in the JOB statements in the RTJOB, JCLJOB,TSOJOB, TSOVVJOB, TSOVFJOB and TSONVJOB members, i.e. no variables are used for the user ID, accounting information and user password. In this case, the specifications apply for all jobs.

These notes also apply to default jobs created by openFT if the TSOJOB, TSOVVJOB, TSOVFJOB and TSONVJOB members do not exist.

The examples below reflect the default structure of the jobs created by openFT. Deviations specific to the computer center can be implemented in the members PRTJOB, JCLJOB, TSOJOB, TSOVVJOB, TSOVFJOB and TSONVJOB.

Example of the member PRTJOB

```
//&JOBP##P JOB &ACCN#########################,
//             &PGRN################,
//             MSGCLASS=X,
//             CLASS=C,
//             USER=&USID###,PASSWORD=&PASS###
//PRTJOB    EXEC PGM=IEBPTPCH
//SYSPRINT  DD DUMMY
//SYSUT1    DD DSN=&RLFN#########################################,
//          DISP=(SHR,DELETE),
//          DCB=(RECFM=FB,LRECL=134,BLKSIZE=2546)
//SYSUT2    DD SYSOUT=A,DCB=(LRECL=134)
//SYSIN     DD *
     PRINT PREFORM=A
/*
//
```

In this example the user ID, account number and password are inserted by openFT from the user's TRANSFER-ADMISSION entry. For the PGRN variable, the value specified by the user with the keyword PGRN= as subcommand in the command string for follow-up processing is used (for more detail, please refer to the User Guide "openFT for z/OS - Managed File Transfer in the Open World"). If no value exists for „programmer's name", this card is ignored, i.e. there is no execution, since it contains no other data except for the JCL identifier „//" and comma (separator for parameters). If job processing is successful, the value is taken from the command string in the SUCCESS-PROCESSING parameter; if not, it is taken from the command string in the FAILURE-PROCESSING parameter. If no value exists for „programmer's name", this card is ignored, i.e. there is no execution, since it contains no other data except for the JCL identifier „//" and comma (separator for parameters). In addition, the name of the file with the result list is inserted (the field &RFLN##...## provided for this name should be 32 characters long). openFT then initiates this job.

Example of the member JCLJOB

```
//&JOBP##N JOB &ACCN#######################,
//         MSGCLASS=X,
//         CLASS=C,
//         REGION=2M,
//         USER=&USID###,
//         PASSWORD=&PASS###
//JOBLIB    DD  DSN=&USID###.PROCLIB,DISP=SHR
```

In this example the user ID, the account number and the password are inserted by openFT from the user's PROCESSING-ADMISSION entry. For the PGRN variable, the value specified by the user with the keyword PGRN= as subcommand in the command string for follow-up processing is used (for more detail, please refer to the User Guide "openFT for z/OS - Managed File Transfer in the Open World"). If no value exists for „programmer's name", this card is ignored, i.e. there is no execution, since it contains no other data except for the JCL identifier „//" and comma (separator for parameters).

The JCL statements specified by the user in the NCOPY command as follow-up processing are added by openFT after the prototype statement "//JOBLIB DD ...". openFT then initiates this job.

Example of the member TSOJOB

```
//&JOBP##N  JOB  &ACCN###################################,
//          &PGRN###############,
//          MSGCLASS=X,
//          CLASS=C,
//          NOTIFY=&USID###,
//          USER=&USID###,
//          PASSWORD=&PASS###,
//          REGION=0M
//          EXEC PGM=IKJEFT01
//OPENFT    DD DSN=&CONN#########################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#########################################,
//             DISP=(SHR,KEEP)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
```

In this example the user ID, the account number and the password are inserted by openFT from the user's PROCESSING-ADMISSION entry. For the PGRN variable, the value specified by the user with the keyword PGRN= as subcommand in the command string for follow-up processing is used (for more detail, please refer to the User Guide "openFT for z/OS - Managed File Transfer in the Open World"). If no value exists for „programmer's name", this card is ignored, i.e. there is no execution, since it contains no other data except for the JCL identifier „//" and comma (separator for parameters). The TSO commands specified by the user in the NCOPY command as follow-up processing, as well as the end of data terminator "/*", are added by openFT after the prototype statement "//SYSTSIN DD *". openFT then initiates this job.

Example of the member TSOVVJOB

```
//&JOBP##N  JOB  &ACCN###################################,
//          MSGCLASS=X,
//          CLASS=C,MSGLEVEL=(1,1),
//          USER=&USID###,
//          PASSWORD=&PASS###,
//          REGION=0M
//STEP0     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#########################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#########################################,
//             DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
FTATTP &PRID
```

In this example, the user ID, account number and password are inserted by openFT from the user’s TRANSFER-ADMISSION. openFT allocates the appropriate files for the variables CONN, NCLO and RLFP.

If %TEMPFILE was not specified during pre-processing, openFT extends the job as follows:

```
//********************************************************
//STEP1     EXEC PGM=IEFBR14
//STDOUT    DD DSN=&RLFP#################################,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(DSORG=PS,BLKSIZE=1536,RECFM=VB),
//          UNIT=SYSDA,SPACE=(1,(256,2560)),AVGREC=K
//********************************************************
//IFBAD     IF STEP0.RC=0 THEN
//STEP2     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#################################,
//          DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#################################,
//          DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD DSN=*.STEP1.STDOUT,DISP=(MOD,KEEP,DELETE)
//SYSTSIN   DD *
//IFBADEND  ENDIF
```

If %TEMPFILE was specified during pre-processing, openFT extends the job as follows:

```
//IFBAD     IF STEP0.RC=0 THEN
//STEP2     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#################################,
//          DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#################################,
//          DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
//IFBADEND  ENDIF
```

Example of pre-processing using %TEMPFILE on a Windows or Unix system:

```
ncopy zospartner!"|ftscopy from,WindowsPC,(%tempfile),*any('hallo.txt'\
,trans='WindowTransadm')" - zosTransadm
```

openFT inserts the TSO or openFT commands specified as preprocessing in the FT request as instream data cards after the template statement "//SYSTSIN DD *". The TSO commands must comply with the IBM conventions. They can be of any length and, if necessary, openFT will spread the command over multiple lines. Any output from the openFT commands as part of preprocessing is redirected to SYSPRINT and consequently to the file referenced by RLFP. By default, TSO commands output to SYSTSPRT. It may be necessary to redirect this output to SYSPRINT and thus to the output file for preprocessing (e.g. LISTCAT OFILE(SYSPRINT)). When the preprocessing commands have been read in, openFT passes the subsequent commands to the internal reader for batch processing (this part is generated dynamically and cannot be modified).

```
//********************************************************
//IFBAD IF (ABEND OR STEP2.RC>=12 OR NOT STEP2.RUN)
//          THEN
//STEP3     EXEC PGM=IKJEFT01
//OPENFT    DD DSN=&CONN#################################,
//          DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#################################,
//          DISP=(SHR,KEEP)
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
FTDETP FAILURE &PRID
//          ELSE
//STEP4     EXEC PGM=IKJEFT01
//OPENFT    DD DSN=&CONN#################################,
//          DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#################################,
//          DISP=(SHR,KEEP)
```

```
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
FTDETP SUCCESSFUL &PRID
//IFBADEND  ENDIF
/*
```

openFT then starts this job. If processing is aborted, openFT starts its own "Cancel-Job" on the basis of the job envelope TSOJOB. This job is assigned the letter "Z" as the last letter of the job name to give it a higher priority than the current processing jobs.

Example of the member TSOVFJOB

```
//&JOBP##N  JOB  &ACCN###################################,
//          MSGCLASS=X,
//          USER=&USID###,
//          NOTIFY=&USID###,
//          PASSWORD=&PASS###,
//          REGION=0M
//STEP0     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#################################,
//             DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
FTATTP &PRID
//********************************************************
//STEP1     EXEC PGM=IEFBR14
//STDOUT    DD DSN=&RLFP#################################,
//             DISP=(NEW,CATLG,DELETE),
//             DCB=(DSORG=PS,BLKSIZE=1536,RECFM=VB),
//             UNIT=SYSDA,SPACE=(1,(256,2560)),AVGREC=K
//STDERR    DD DSN=&RLFT#################################,
//             DISP=(NEW,CATLG,DELETE),
//             DCB=(DSORG=PS,BLKSIZE=1536,RECFM=VB,LRECL=1532),
//             UNIT=SYSDA,SPACE=(1,(256,2560)),AVGREC=K
//SYSERR    DD DSN=&RLFF#################################,
//             DISP=(NEW,CATLG,DELETE),
//             DCB=(DSORG=PS,BLKSIZE=1536,RECFM=VB,LRECL=1532),
//             UNIT=SYSDA,SPACE=(1,(256,2560)),AVGREC=K
//********************************************************
//IFBAD     IF STEP0.RC=0 THEN
//STEP2     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#################################,
//              DISP=(SHR,KEEP)
//STEPLIB    DD DSN=&NCLO#################################,
//              DISP=(SHR,KEEP)
//SYSPRINT   DD DSN=*.STEP1.STDOUT,DISP=(MOD,KEEP,DELETE)
//SYSTSPRT   DD DSN=*.STEP1.STDERR,DISP=(MOD,KEEP,DELETE)
//SYSERR     DD DSN=*.STEP1.SYSERR,DISP=(MOD,KEEP,DELETE)
//SYSTSIN    DD *
```

A special form of preprocessing in z/OS takes the form of the server function for an "ftexec" command issued in partner system. ftexec expects to be returned the output from the passed commands (stdout), any error messages that occur (stderr) and an exit code.

openFT inserts the TSO or openFT commands specified as preprocessing in ftexec as instream data cards after the template statement "//SYSTSIN DD *". The TSO commands must comply with the IBM conventions. They can be of any length and, if necessary, openFT will spread the command over multiple lines. openFT dynamically appends the output from SYSTSPRT to that of SYSERR. To do this, it internally uses the IBM utility IEBGENER:

```
//IFBADEND   ENDIF
//STEP22     EXEC PGM=IEBGENER
//SYSUT1     DD DSN=*.STEP2.SYSTSPRT,DISP=(MOD,DELETE,DELETE)
//SYSUT2     DD DSN=*.STEP2.SYSERR,DISP=(MOD,KEEP,DELETE)
//SYSIN      DD DUMMY
//SYSPRINT   DD SYSOUT=*
//SYSTSIN    DD *
```

The content of the temporary file generated by this is redirected to "stderr" in the partner system and SYSPRINT is redirected to "stdout". After reading in the commands, openFT extends the job in accordance with the example for TSOVVJOB, see job steps STEP 3 and STEP 4 on page 69.

Example for the member TSONVJOB

```
//&JOBP##N  JOB  &ACCN###################################,
//          MSGCLASS=X,
//          CLASS=C,MSGLEVEL=(1,1),
//          USER=&USID###,
//          PASSWORD=&PASS###,
//          REGION=0M
//STEP1     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#########################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#########################################,
//             DISP=(SHR,KEEP)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD *
FTATTP &PRID
//********************************************************
//IFBAD     IF STEP1.RC=0 THEN
//STEP2     EXEC PGM=IKJEFT01,
//          COND=(0,NE)
//OPENFT    DD DSN=&CONN#########################################,
//             DISP=(SHR,KEEP)
//STEPLIB   DD DSN=&NCLO#########################################,
//            DISP=(SHR,KEEP)
```

In the case of postprocessing in z/OS, the transferred data is first stored in a temporary file which is then available as input for the commands specified in the request (TSO commands or system commands for corresponding utilities). The temporary file can be directly referenced in the commands by means of the metastring %TEMPFILE.

openFT then extends the job as follows:

```
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
//SYSTSIN   DD *
//IFBADEND  ENDIF
```

If %TEMPFILE is not used in the postprocessing commands, then postprocessing reads in the data via SYSUT1. The data stream from the FT request read via SYSTSIN then uses the data specified in SYSUT1 as input data. To this end, users can specify a corresponding utility program in the postprocessing operation in order to access this data.

openFT then extends the job as follows:

```
//SYSUT1    D     DSN=&RLFP#########################################,
//                  DISP=(MOD,KEEP,DELETE)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
//SYSTSIN   DD *
//IFBADEND  ENDIF
```

Example

The user wants to copy a file DAT1 from a Windows or Unix system to z/OS using ncopy. Via the postprocessing operation, the file DAT1 is to copy to another file DAT2 in the target system. The utility IEBGENER is used:

```
ncopy DAT1 ZosPartner!”|allocate dsname(DAT2) ddname(SYSUT2) mod keep
dsorg(ps) recfm(v,b) lrecl(259); ‘call SYS1.LINLLIB(IEBGENER)’”
UserId,Account,Password
```

In all cases, the postprocessing job is extended by further statements after the processing commands in the same way as preprocessing jobs. This is performed in the same way as for the preprocessing job (see TSOVVJOB on page 68).

If processing is aborted, openFT starts its own "Cancel-Job" on the basis of the job envelope TSOJOB. This job is assigned the letter "Z" as the last letter of the job name to give it a higher priority than the current processing jobs, see TSOVVJOB with the job steps STEP 3 and STEP 4 on page 69.

2.6.2.4 Structure of the members SUCCMSG and FAILMSG

These members contain the installation-specific texts issued by openFT as a result of successful or unsuccessful file transfer to one or several consoles.

These members are evaluated only if a valid specification has been made for the SUCC_MSG or FAIL_MSG keyword in the FTMSPPAR member and if ENDMSG_ROUTCDE=1 has been set.

If the FTMSPPAR member contains valid specifications for SUCC_MSG or FAIL_MSG, and the members SUCCMSG and FAILMSG exist but are empty, no message is output.

The following rules apply to the installation-specific message texts:

- The text must begin and end with single quotes. These characters are not output but form part of the syntax.
- After replacement of the variables (see below), the text may be up to 102 characters long, **including** the single quotes in which it is enclosed.
- Like all other members of the FT parameter library, the members containing the message texts may not contain line numbering.

i A single quote within a message no longer has to be duplicated as was the case in earlier versions. If you have already analyzed these messages prior to migration to V11, for instance using NetView, you should retain the duplicated quotes for reasons of compatibility.

Violation of these rules results in an **error**. At worst, either no message is output at all or a message is issued to **all** TSO terminals currently active.

Like the members PRTJOB, JCLJOB and TSOJOB, the message texts can contain variables which openFT replaces with the current values.

The following variables may be used in message texts:

FILX

Name of the send or receive file as specified in the NCOPY command for the system involved. Maximum 58 characters in accordance with IBM conventions.

When replacing this variable with the current value, openFT duplicates single quotes enclosing fully qualified file names, thus satisfying the syntax rule "If a single quote occurs in the message itself, it must be duplicated". (Single quotes are not duplicated in the message itself.) The maximum length of 58 characters for this parameter is formed as follows: 44 characters (maximum length of a fully qualified file name, not including the single quotes which enclose it) + 8 characters (maximum length of a member name) + 2 characters (parentheses enclosing the member name) + 4 (2 x 2 single quotes).

PNAM

Symbolic name of the remote system. Maximum 8 characters in accordance with openFT conventions.

If there is no symbolic partner name for this request, the first 8 characters of the partner address are output.

SUBM

specifies the system in which the FT request was submitted. Maximum 6 characters. openFT replaces this variable with the following character strings:

LOCAL	if the request was submitted in the local system,
REMOTE	if the request was submitted in a remote system.

USID

User ID from TRANSFER-ADMISSION.
Maximum 7 characters in accordance with IBM conventions.

ACCX

"accounting information" from the TRANSFER-ADMISSION.
Maximum 42 characters in accordance with IBM conventions.

When replacing this variable with the current value, openFT duplicates single quotes that can enclose "accounting information" (see section "Access authorization" in the User Guide openFT for z/OS - Managed File Transfer in the Open World"). This satisfies the syntax rule If there is a quote, use double quotes (only single quotes appear in the message itself).

The maximum length of 42 characters for this parameter is formed as follows: 40 characters for the "accounting information" + 2 additional quotes.

OWID

Owner of the FT request (user ID under which the FT request was submitted). Maximum 7 characters in accordance with IBM conventions. This variable is replaced with a valid value only for FT requests which were submitted in the local system. This variable is eliminated for FT requests submitted in a remote system.

PGRX

The programmer's name as specified as a command prefix with the key PGRN= in the relevant command string for follow-up processing; i.e. in the member SUCCMSG, the value is taken from the command string in the parameter SUCCESS-PROCESSING, in the member FAILMSG it is taken from the command string in the parameter FAILURE-PROCESSING. For further details refer to the user manual "openFT for z/OS - Managed File Transfer in the Open World". Up to 20 characters, according to IBM conventions, plus the number of apostrophes possibly enclosing or contained in the current value.

When replacing the variable by the current value, openFT doubles the number of apostrophes that can enclose or be contained in a programmer's name. This fulfills the syntax rule "If an apostrophe occurs in a message, then double it". (Only single apostrophes appear in the message itself.)

This is also why the length of the field has to be increased by the number of apostrophes possibly enclosing or contained in the current value.

TRID

FT transfer identification. Maximum 10 characters (value range 1..2147483639) in accordance with openFT conventions.

The name of this variable must be given the prefix "%" or - for reasons of compatibility with predecessor versions - „&".

As many "#" fill characters as necessary should be appended to the names of these variables so that the field length reaches the maximum length (including the "&", for example &PNAM###). If openFT replaces the variables with the current values, it does not exceed the field length defined by the name of the variables including the fill characters; if necessary, the current values are truncated. Syntax errors caused by truncation can also lead to a **misfunction**.

In the opposite direction, openFT removes superfluous fill characters from current values that are shorter than these field lengths.

Example of the member SUCCMSG

(The FAILMSG member must have the same structure)

```
'DATASET␣&FILX####.....####␣TRANSFERRED␣
TO/FROM␣&PNAM###.'
```

The field &FILX####.....#### for the file name should have a total length of 58 characters. The first line of the member is 80 characters long (including the single quote at the start); the rest of the message (including the single quote at the end) is located on the second line. The total length of the message (maximum 98 characters including the single quotes) does not exceed the maximum permitted value of 102 characters.

If the NCOPY command was entered as follows:

```
NCOPY TRANS=TO,
      PARTNER=SYS1,
      LOC=(FILE='USER1.ABC',TRANS=(USER2,ACC2,PASS2)),
      REM=...
```

then the following message is issued in this example after successful file transfer:

```
DATASET 'USER1.ABC' TRANSFERRED TO/FROM SYS1. CN(00)
```

2.6.2.5 Structure of the member TNSTCPIP

This member contains address entries (transport system addresses) for partner systems which are to be accessed via TCP/IP. openFT for z/OS accesses these address entries via the entry for the relevant partner system in the partner list.

> i This member is no longer required as of V10, as all TCP/IP partners can be addressed without TNSTCPIP entries. This applies to both partners from the partner list and to dynamic partners. TNSTCPIP is only supported for reasons of compatibility, in other words, TNSTCPIP entries can be taken over from an older version of openFT.

A maximum of 10000 records in the TNSTCPIP member are evaluated.

Each record must contain precisely one entry of the following form:

```
tns-name=internet-address:[port-number]:tsel:[comment]
```

Meaning:

tns-name

Name of the address entry. This name must be specified as a component of the PARTNER-ADDRESS= parameter in the FTADDPTN command (see page 195) for a partner system which is to be accessed via TCP/IP.

The local openFT instance must use a unique name for the tns-name of each openFT partner system.

tns-name may consist of up to 8 alphanumeric characters, the first of which must be a letter or one of the special characters $, @ or #.

internet-address

Internet address of the remote computer (openFT partner system). The Internet address can be specified in one of the following formats:

– in the format xxx.xxx.xxx.xxx where xxx stands for an integer (decimal presentation) in the range from 0 to 255,

– as a symbolic name through which openFT can determine the Internet address of the remote system using the function GETHOSTBYNAME. (The function GETHOSTBYNAME can only be called if the C runtime system for TCP/IP is installed; the function supplies the address belonging to the symbolic name, if it can be determined using the TSO command NSLOOKUP.)

port-number

Port number of the openFT instance on the remote system (openFT partner system). The port number is an integer in the range 1 to 32767.

By default, all openFT partners use port number 1100. If different settings apply in the partner system then the corresponding value must be entered here.

tsel

T-selector (TSEL) of the openFT instance on the remote system (openFT partner system). The T-selector can consist of up to 32 characters.

For partner systems using the recommended settings or on which the installation settings were not modified, $FJAM␣␣␣ (openFT partner) is specified here, otherwise the T-selector specified in the partner system.

comment

Any characters preceded by three colons on the same line are ignored. This field can consequently be used as a comment field.

Example of the TNSTCPIP member

```
XAS1=149.202.138.246:1100:$FJAM  : ENTRY F. XAS1 (openFT f. z/OS)
JUMBO=149.202.138.245::$FJAM   : ENTRY F. JUMBO (openFT f. BS2000)
SYS768=149.202.138.84::$FJAM   : ENTRY F. SYS768 (openFT f. Unix Systems w. CMX)
SYS123=sys123.xxxx.yyy.de:1100:$FJAM  : ENTRY WITH A SYMBOLIC NAME
```

This example lists entries in the TNSTCPIP member of an openFT system which is running without extended authentication for the following partner systems:

XAS1

Entry for a remote openFT for z/OS system which is connected to the local openFT instance directly via TCP/IP.

The Internet address of the remote computer is 149.202.138.246. Port number 1100 was assigned to the main station of the remote z/OS partner system using the openFT operating parameter OPENFT-APPL (default setting). PARTNER-ADDRESS=XAS1 must be specified in the FTADDPTN command used to enter this remote z/OS partner system in the partner list of the local openFT instance.

JUMBO

Entry for a remote openFT for BS2000 system which is connected to the local openFT instance directly via TCP/IP. The Internet address of the remote computer is 149.202.138.245 and the port number is 1100.
PARTNER-ADDRESS=JUMBO must be specified in the FTADDPTN command used to enter this remote BS2000 partner system in the partner list of the local openFT instance.

SYS768

Entry for a remote Unix system with openFT which is connected to the local openFT instance directly via TCP/IP. The Internet address of the remote computer is 149.202.138.84. CMX V6.0 is installed on the Unix computer. The T-selector $FJAM was assigned to the main station of the Unix partner system using the tnsxcom statement

```
$FJAM\
      TSEL RFC1006 T'$FJAM'
      TSEL LANINET A'1100'
```

PARTNER-ADDRESS=SYS768 must be specified in the FTADDPTN command used to enter this remote Unix partner system in the partner list of the local openFT instance.

SYS123

Entry for a remote system of the type openFT for z/OS which is linked directly with the local openFT instance via TCP/IP.
openFT can determine the Internet address of the remote system through the symbolic name sys123.xxxx.yyy.de using the function GETHOSTBYNAME.
The main station of the z/OS partner system was allocated to port 1100 using the openFT operating parameter OPENFT-APPL (default setting). PARTNER-ADDRESS=SYS123 must be specified in the FTADDPTN command with which this z/OS partner system is entered in the partner list of the local openFT instance.

2.6.2.6 Structure of the member FNAMECTB

This element contains information on which file-specific character sets openFT is to use and on which files are to be encoded with which character sets. A range of character sets in the form of code tables are supplied with openFT. See also section “Administering code tables” on page 103.

The character set is selected by means of the name of the send or receive file. If openFT-AC is used, the file name may consist of the specification from the transfer request and from the admission profile accessed in the transfer request.

Each line of the member must contain one of the following specifications:

- the name of the character set (code table) in the following format:

  ```
  @ctabname          [comment]
  ```

  ctabname is the name of the code table (1 to 8 characters, also known as the CCS name).

- a selection pattern for selecting file names where the "*" character can be used as a placeholder for a part of the file name; the "*" character may only be used once in a selection pattern. It stands for an optional number of characters (including 0 characters).

The following rules must be observed when creating the member FNAMECTB:

- There must be no blank between "@" and the name of the character set.
- A line containing the name of a character set may be followed by one or more lines with selection patterns for file names. All files whose names match one of the selection patterns are allocated to that character set.
- A table name to which no selection pattern is allocated is skipped.
- If a table name is specified several times, the character set is loaded several times; all specified allocations to selection patterns are considered.
- If a selection pattern is specified more than once, the first allocation applies.
- If a file name matches several selection patterns, the first hit applies. More specific selection patterns must therefore be located before a more general selection pattern.
- Leading blanks in a line are ignored so that the allocation specifications can be clearly structured.
- All lines up until the first occurrence of a table name are treated as comment lines.
- At the end of each line, a comment can be included, separated by at least one blank from the rest of the line.
- File name patterns not enclosed in single quotes ignore the first-level qualifier if the files are not openEdition files.

Example for the pattern *A.TEXT:

'USERA.TEXT'	does not match, because the A is not part of the first-level qualifier
'USER1.AAA.TEXT'	matches
/AAA.TEXT	matches, because it is an openEdition file

openFT assigns a table ID to each code table that is loaded. A list of all code tables is entered into the openFT job log; the table IDs and the allocated selection patterns for file names are also listed there for each code table. Incorrect specifications in FNAMECTB and errors that occur when a table is loaded are indicated by means of a negative number as table ID. The individual values have the following meanings:

Table ID	**Meaning**
-1	Syntax error in the selection pattern specification
-2	Code table could not be loaded
-3	Syntax error in the specification of the name of a code table, for instance a blank between the "@" and the name of the code table or the name is too long (in this case the first 8 characters of the name are shown)

Errors that occur when the member FNAMECTB is read in are not logged; the allocation list that was created up until the first error occurred is used.

Example for the member FNAMECTB

In this example, the possible entries (including some erroneous entries) in the member FNAMECTB are presented along with effect and including the entries in the openFT job log which they generate.

```
THIS MEMBER DEFINES FILE SPECIFIC CODE TABLES FOR OPENFT

@FNCOD001         - SPECIAL TABLE 1
   'USER1.*.TEXT'
@FNCOD002         - SPECIAL TABLE 2
   'USER2.ABC*'
   'USER2.*ABC'
   *TEXT*
   'USER3.DEF'
@IBM037         - EBCDIC CODE PAGE 037
   *.TEXT
@IBM273          - EBCDIC CODE PAGE 273
   DEF
@IBM500          - EBCDIC CODE PAGE 500
   *.CHAR
@FNCODTABL
   XYZ*
@ FNCOD000
   *CHAR
@UTF8
   *.tst
```

The resulting entries in the openFT job log:

```
TABLE_ID CODETABLE PATTERN
0014FE00 FNCOD001  'USER1.*.TEXT'                                > (1)
0014FC00 FNCOD002  'USER2.ABC*'                                  > (2)
0014FC00 FNCOD002  'USER2.*ABC'                                  > (3)
-1       FNCOD002  *TEXT*                                        > (4)
0014FC00 FNCOD002  'USER3.DEF'                                   > (5)
0014FA00 IBM037    *.TEXT                                        > (6)
-2       IBM273    DEF                                           > (7)
0014F800 IBM500    *.CHAR                                        > (8)
-3       FNCODTAB  XYZ*                                          > (9)
-3       FNCOD00   *CHAR                                         > (10)
001E445F  UTF8     *.tst                                         > (11)
```

Explanations

(1) All files with the first level qualifier USER1 whose names end with ".TEXT" are coded using the character set FNCOD001.

(2) All files with the first level qualifier USER2 whose partially qualified names begin with "ABC" are coded using the character set FNCOD002.

(3) All files with the first level qualifier USER2 whose names end with "ABC" are also coded using the character set FNCOD002.

(4) Syntax errors in the specified selection patterns: the "*" character is used several times.

(5) The file 'USER3.DEF' is also coded using the character set FNCOD002.

(6) All files whose names end with ".TEXT" are coded using the character set IBM037 (except those whose first level qualifier is USER1, see (1)).

(7) All files with a partially qualified name DEF (except 'USER3.DEF' - see (5)) are to be are coded using the character set IBM273, but this table cannot be loaded, for instance because it has been deleted from the library.

(8) All files whose names end with ".CHAR" are coded using the character set IBM500.

(9) Syntax errors in the specification of the name of the code table: Name longer than 8 characters. Note: This entry is to capture all files whose partially qualified name begins with "ABC" but with the exception of the files whose first level qualifier is USER2 - see (2).

(10) Syntax errors in the specification of the name of the code table: There is a blank between "@" and the name of the code table. - Note: This entry is to capture all files whose names end with "CHAR" with the exception of the files whose names end with ".CHAR" - see (8).

(11) All files with names ending in ".tst" are encoded using the UTF8 character set.

2.6.2.7 Structure of the member FTACPAR

The installation parameters which are needed when openFT-AC is used are stored in this member.

When the openFT load module is started for the first time after the delivery unit openFT-AC is installed, openFT automatically generates the FTAC file (see section “Internal openFT data sets” on page 424) using the characteristics specified for its name and size stored in this member or the default values.

Each line of the FTACPAR member can contain exactly one parameter in the form "keyword=value". No blanks may be inserted between "keyword", "=" and "value". Below is a list of the keywords which may be used.

Keywords:

FILE_2ND_Q=

The second level qualifier for the name of the components of the FTAC file (see section “Internal openFT data sets” on page 424).
Up to 17 characters (default: <inst>.SYSFSA, where <inst> is the name of the openFT instance). For the sake of clarity, the name should always start with the instance name followed by a period.

FILE_SIZE_KB=

Initial size of the FTAC file (in KB).
Maximum value: 30736382 (default: 1024).

openFT uses this value as the primary allocation size when creating the VSAM cluster which is part of the FTAC file. The value is halved for the size of the secondary allocation.

The specified maximum value is the program-technical limit. When choosing a value for the initial size of the FTAC file, the actually available storage space needs to be taken into consideration. Note that the FTAC file is created on the same data volume as the request file, the partner list, the operating parameter file and the logging file (see section “Internal openFT data sets” on page 424).

Example for the FTACPAR member

```
FILE_2ND_Q=HAPPI.OPENFTAC
```

2.6.3 Providing the OPFT subsystem

The commands in the dialog tasks with which FT users and administrators work (NCOPY or an alias, or FTHELP or FTTRACE) are encrypted for the purposes of internal communication with openFT. This encryption (and decryption) is performed by the OPFT subsystem. In addition, OPFT administers the running openFT instances. Optionally, it is also possible to use the MSG_CRYPT parameter in the PARM library to activate message encryption. OPFT must be installed in the computer's IPL. To do this, it is necessary to copy the members from OPENFT.LPALIB to SYS1.LPALIB or store them in a user LPALIB that is concatenated with this library.

- IGX00211 (SVC handler)
- OPFTIGX
- OPFTINIT (Startup routine of the OPFT subsystem)
- OPFTSUB (Subsystem handler)

It is important to avoid name conflicts with load modules that already exist in SYS1.LPALIB.

The portal to the subsystem is implemented via SVC 109 with "extended code 211". If "extended code 211" is already used for a different purpose in your system then the samples in openFT provide a procedure with the name LINKIGX that allows you to generate other, alternative "extended codes" that can be set in the PARM file with 'OPENFT_SVC='.

The subsystem is initialized the first time openFT is started after IPL. The start of the subsystem is confirmed by a console message. After initialization, the subsystem remains active until the next IPL. No further administration is necessary and, in particular, the subsystem does not require any start parameters.

2.6.4 openFT as a job or started task

The OPENFT load module runs either as an ordinary batch job or as a started task. In both cases, the associated user ID must possess the authorizations described in the section “openFT privileges” on page 31.

The FJGEN command (see page 48) generates JCL statements for loading a batch job and starting the openFT load module. This JCL is entered in the FJBATCH member of the FT procedure library <openft qualifier>.<inst>.CLIST.

Example of the FJBATCH member

```
//OPENFTF JOB (A123,B123),
//          CLASS=A,MSGCLASS=A,
//          USER=OPENFT,PASSWORD=OPENFT,
//          TIME=1440,REGION=0M
//DLTDMP    EXEC PGM=IEFBR14
//DELFILE   DD OPENFTQU.STD.SYSUDUMP,
//             DISP=(MOD,DELETE,DELETE),
//             SPACE=(CYL,(20,5)),
//             DCB=(DSORG=PS)
//OPENFT    EXEC PGM=OPENFT,TIME=1440,
//          PARM='OPENFTQU,VSN123/SYSDA,A,FTID1,STD,AFFE,1100,'
//*            openFT V11.0A00 / FJBATCH V11.0A00
//STEPLIB   DD DSNAME=OPENFT.OPENFT.LOAD,
//             DISP=(SHR,KEEP)
//OPENFTS   DD DSNAME=OPENFT.OPENFT.NCLOAD,
//             DISP=(SHR,KEEP)
//OPENFT    DD DSNAME=OPENFTQU.STD.CONN,
//             DISP=(SHR,KEEP)
//OPFTATT   DD DSNAME=OPENFTQU.STD.OPFTATT,
//             DISP=(SHR,KEEP)
//*DDUADS    DD DSNAME=SYS1.UADS,
//*             DISP=(SHR,KEEP)
//OPFTHSM   DD DSNAME=OPENFTQU.STD.COLLECT.DATA,
//             DISP=(SHR,KEEP)
//MCDS      DD DSNAME=DFHSM.MCDS,DISP=SHR
//SYSIN     DD DUMMY
//SYSOUT    DD DUMMY
//IEBCOUT   DD DUMMY
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD DSN=OPENFTQU.STD.SYSUDUMP,
//             SPACE=(CYL,(20,5)),DISP=(,CATLG),
//             DCB=(DSORG=PS)
```

The JCL card for SYS1.UADS has been commented out as it is not required when RACF is used.

The DD cards OPFTHSM and MCDS are required for archiving and retrieving files.

FJGEN creates the FBATCH member using the specified installation parameters (for further information, see the description of the FJGEN command, page 182).

You can adapt the JCL statements in this procedure to meet the requirements of your installation.

For example, if a file named SYS1.UADS exists on your system whereas validation of the user ID is to be performed via RACF then you must remove the two lines with the assignment of DDUADS to SYS1.UADS from the procedure.

In the statement `//OPENFT    EXEC PGM=OPENFT, ...` you can replace the program name OPENFT with the aliases OPENFTS or OPENFTSL described in section “Reading in the openFT product tape” on page 36, see also page 37.

If a local host name other than the default has been specified for the openFT instance in FJGEN then this is entered after the port number in the PARM parameter of the //OPENFT ... statement.

You can redirect the openFT job log to a file by modifying the DD statement with the label `SYSPRINT`. Attention must be paid to the following factors:

- Output can be directed into a PS data set or into a PO or PDSE member. However, an existing PO or PDSE member cannot be extended.
- If the file is to be newly created, do not make any specifications concerning the record length (LRECL) and the block size (BLKSIZE). openFT generates the file with LRECL=1536 and BLKSIZE=1536.
- If you are using an existing file (i.e. if you are extending a PS data set or if you are creating a new member in an existing PO or PDSE data set), the file must have the attributes LRECL=512 and BLKSIZE=512.

Examples

- If the PS data set does not yet exist, it is to be created; otherwise it is to be extended:

```
//SYSPRINT DD DSN=USERID.LOG1,DISP=(MOD,CATLG),RECFM=FB,
//            SPACE=(CYL,(20,20))
```

- An existing PS data set is to be overwritten:

```
//SYSPRINT DD DSN=USERID.LOG2,DISP=(OLD)
```

- The PO or PDSE data set already exists. If the member does not yet exist, it is to be created; otherwise it is to be overwritten:

```
//SYSPRINT DD DSN=USERID.LOG3(MEMBER1),DISP=(OLD)
```

The DD statement with the label `SYSUDUMP` causes openFT to write the dump to this file in printable form on a "Cancel with Dump". Other system dumps are output to SYSFDF.

If openFT is to run as a started task, which means that it is to be started either automatically when the system starts or by means of an operator command, a specific start procedure must be created by the user's computer center. The FJBATCH created with the FJGEN can be used as a template to be copied.

Example of a start procedure:

```
//           PROC
//DLTDMP     EXEC PGM=IEFBR14
//DELFILE    DD OPENFTQU.STD.SYSUDUMP,
//              DISP=(MOD,DELETE,DELETE),
//              SPACE=(CYL,(20,5)),
//              DCB=(DSORG=PS)
//OPENFT     EXEC PGM=OPENFT,TIME=1440,
//           PARM='OPENFTQU,VSN123/SYSDA,A,FTID1,STD,AFFE,1100,'
//*            openFT V11.0A00 / FJBATCH V11.0A00
//STEPLIB    DD DSNAME=OPENFT.OPENFT.LOAD,
//              DISP=(SHR,KEEP)
//OPENFTS    DD DSNAME=OPENFT.OPENFT.NCLOAD,
//              DISP=(SHR,KEEP)
//OPENFT     DD DSNAME=OPENFTQU.STD.CONN,
//              DISP=(SHR,KEEP)
//OPFTATT    DD DSNAME=OPENFTQU.STD.OPFTATT,
//              DISP=(SHR,KEEP)
//OPFTHSM    DD DSNAME=OPENFTQU.STD.COLLECT.DATA,
//              DISP=(SHR,KEEP)
//MCDS       DD DSNAME=DFHSM.MCDS,DISP=SHR
//SYSIN      DD DUMMY
//SYSOUT     DD DUMMY
//IEBCOUT    DD DUMMY
//SYSPRINT   DD SYSOUT=*
//SYSUDUMP   DD DSN=OPENFTQU.STD.SYSUDUMP,
//              SPACE=(CYL,(20,5)),DISP=(,CATLG),
//              DCB=(DSORG=PS)
```

The installation parameters described in the section the FJGEN command (see page 182) must also be used in this start procedure.

The explanations given above for adapting the FJBATCH member (program name OPENFT, SYSPRINT, SYSUDUMP) also apply here.

2.6.5 Loading and starting the openFT load module

By using the FJGEN command and the entries associated with it a batch job was created in the openFT instance's FT procedure library in the member FJBATCH. This job can now be executed with the FJINIT command.

The member FJBATCH can be adapted to the conventions of your computer center if the batch job does not comply with them.

Alternatively, openFT can also be started as a started task.

For more details, see the description of the commands FJGEN (page 182) and FJINIT (page 193) as well as the section "openFT as a job or started task" on page 85.

2.6.6 Activating, deactivating and terminating openFT

After the openFT load module has been loaded, openFT can be activated using the FTSTART command.

File transfer requests cannot be accepted or executed until openFT has been activated.

openFT is deactivated using the FTSTOP command. Once this command has been issued, no more file transfer requests can be executed. openFT is terminated using the FTTERM command. If necessary, this command deactivates openFT (FTSTOP command) and terminates the openFT job.

2.7 Linking openFT with data protection products

For each file transfer request and file management request, openFT checks

- the user's access authorization to the system (transfer admission)
- the user's access authorization to the relevant file
- if preprocessing, postprocessing or follow-up processing is to be triggered following a file transfer request: the user's authorization to do so.

Users must demonstrate their authorization by means of the specifications they make in the TRANSFER-ADMISSION and PROCESSING-ADMISSION operands for the system involved. Transfer requests in which authorization is not demonstrated satisfactorily are rejected.

If FTAC is not used, the user must make the entries required for checking his transfer admission directly in TRANSFER-ADMISSION or PROCESSING-ADMISSION (i.e. LOGON ID consisting of user ID, account number and password). If FTAC is used, a TRANSFER-ADMISSION defined in an admission profile can be specified instead of the LOGON ID. FTAC will then read the information needed for the admission check from the relevant profile (i.e. the LOGON ID consisting of user ID, account number and password).

openFT checks the user's transfer admission using RACF calls or against the entries in the MVS system file SYS1.UADS. Transfer admission can also be checked using RACF calls or by calling the PROTECT macro (for more information, see below). To this end, openFT must be assigned APF authorization (see the section “openFT privileges” on page 31) or read access to SYS1.UADS. openFT does not have write access to SYS1.UADS or to RACF lists.

Since all RACF calls are handled by the RACROUTE macro, it is possible to connect an installation-specific MVS exit routine to the **MVS Router exit** or to use an RACF-compatible software product such as ACF-2 or TOP-SECRET (If TOP-SECRET is used, openFT identifies itself to TOP-SECRET as "OSFSUBT", i.e. "PGM=OSF" must be specified).

Information on the requirements which must be met by an RACF-compatible software product in order to enable openFT to perform system and data access control via this product is given in the product-specific manuals.

The interface of the MVS Router exit is described in the IBM manual "System Programming Library: Resource Access Control Facility (RACF)".

openFT accesses the file SYS1.UADS via the DD name DDUADS (see the corresponding DD statements in the examples in the section “openFT as a job or started task” on page 85). openFT checks whether the file SYS1.UADS is available; this check is carried out only during processing of the first transfer request after loading and starting the openFT load module. If this file is not available (DD statement is missing, file is not available or not readable, etc.), openFT no longer accesses the SYS1.UADS file until termination of the openFT load module. During the processing of all subsequent transfer requests, the SYS1.UADS file is considered to be unavailable.

Notes

- If the transfer request is rejected during synchronous command processing, the NCOPY command is terminated with the return code X'0C'. This also applies to the NCOPY program interface.
- Whether or not follow-up processing takes place following rejection of a transfer request (FAILURE-PROCESSING) depends on which FT system rejects the transfer request:
  - If the transfer request is rejected by the **local** openFT instance, no follow-up processing takes place in either of the two FT systems involved.
  - If the transfer request is rejected in the **remote** system, no follow-up processing takes place in the remote system. In this case, the follow-up processing for unsuccessful file transfer (FAILURE-PROCESSING) is initiated in the local system.

  The message that is issued (e.g. FTR2047, FTR2169) indicates whether the local or remote system rejected the transfer request.

2.7.1 Checking the transfer admission

When FTAC is used, the transfer admission check for file transfer and file management requests is carried out when the request is checked by FTAC, independent of whether the request contains a LOGON ID (user ID, account number, password) or a TRANSFER-ADMISSION defined in a user profile. In the latter case, FTAC reads the information required for the check (i.e. the LOGON ID consisting of user ID, account number, password) from the relevant profile. Like openFT, FTAC performs the transfer admission check using RACF calls or compares the entries with the information contained in the SYS1.UADS system file.

If FTAC is not used, openFT itself checks the transfer admission using the LOGON ID (user ID, account number, password) contained in the request.

The user must provide evidence of his or her transfer admission in TRANSFER-ADMISSION. The check sequence for transfer admission (TRANSFER-ADMISSION) is the same as for checking the admission for follow-up processing (PROCESSING-ADMISSION, see page 95); so both cases are treated the same.

Checking the user ID and password

openFT first uses the RACROUTE macro to check whether the user ID specified in the NCOPY command for the TRANSFER-ADMISSION or the PROCESSING-ADMISSION is valid and whether the associated user password, if any, is correct. (The RACROUTE macro makes use of the RACF macro RACINIT for this purpose.) If the result of this check is negative, the transfer request is rejected and an error message is issued.

If the return code from the RACROUTE macro indicates that neither RACF nor a compatible product (ACF-2, TOP-SECRET) is installed and active, openFT attempts to check the user ID and the associated password using the SYS1.UADS data set.

If the SYS1.UADS data set is also unavailable, no transfer request is processed and an error message is issued.

Checking the account number

openFT takes the account number from the user's specification in the NCOPY parameter ACCOUNT (TRANSFER-ADMISSION or PROCESSING-ADMISSION). If "accounting information" is specified here, openFT extracts the account number from this information. Any "(" and/or " ' " characters at the start of this specification are removed. The string is then searched for the first comma ",". If a comma is found, all characters **preceding** this comma are interpreted as the account number. If, however, one of the characters ")" or " ' " is found first, all characters **preceding** this character are interpreted as the account number. If none of the characters "," or ")" or " ' " are found, the entire string is interpreted as the account number.

If the SYS1.UADS data set is available, openFT checks the account number against the entry which matches to the user ID in this file. If the account number is not entered here, the transfer request is rejected and an error message is issued. If the SYS1.UADS data set is available and no account number is specified in the corresponding operand of the NCOPY command (TRANSFER-ADMISSION or PROCESSING-ADMISSION), **no** check is performed on the account number.

If the SYS1.UADS data set is not available, openFT checks whether RACF (or compatible product) is active and whether the RACF resource class ACCTNUM is active. If this is the case, openFT checks the account number using RACF. If an account number is specified in the corresponding operand of the NCOPY command (TRANSFER-ADMISSION or PROCESSING-ADMISSION), this is used for checking purposes. However, if no account number is specified, openFT looks for the "TSO default account number" in the "TSO segment" of the user-specific data relating to the user ID in the ACF database. (The RACROUTE macro uses the RACF macro RACXRTR for this purpose). If this database contains a value with a maximum length of 40 characters, it is used for checking purposes. However, if it is still not possible to find an account number specification, a "pseudo account number" consisting of 40 "@" characters is used.

The RACROUTE macro uses the RACF macro RACHECK to perform an RACF check of the account number.

If the SYS1.UADS data set is not available and RACF is active, but the RACF resource class ACCTNUM is not active, no default account number is allocated and no account number check is performed.

If the SYS1.UADS data set is not available and RACF is not active, then the transfer request was rejected with an error message when the user ID and password were checked (see above).

A computer center can thus ensure that transfer requests are processed without the user having to specify an account number in the NCOPY command. To this end, the following steps must be taken:

- SYS1.UADS data set available: no further steps necessary.
- SYS1.UADS data set not available, RACF resource class ACCTNUM inactive no further steps necessary
- SYS1.UADS data set not available, RACF resource class ACCTNUM active:

  In this case there are two possibilities:

  - If omission of the account number will cause openFT to use the default account number of the user ID specified in TRANSFER-ADMISSION or PROCESSING-ADMISSION, then no further steps are necessary. The RACF database, must contain a default account number for each relevant user ID.
  - If no default account numbers are used and you do not want openFT to check the account numbers of selected users, you must ensure that the above-mentioned "pseudo account number" (40 "@" characters) is entered in RACF (resource class ACCTNUM), and that only these selected users are authorized to use this "pseudo account number". These users may not then make any specification in the ACCOUNT parameter of the NCOPY command.

2.7.2 Checking access authorization

Access authorization for the file accessed by the request is always performed by openFT itself, regardless of whether FTAC is used. The access authorization is checked after a positive transfer admission check (see previous section). The access authorization is checked for the user named in the request's TRANSFER-ADMISSION or for the user determined by FTAC using the information in the TRANSFER-ADMISSION (in the relevant profile) or by the openFT-specific exit routine

The procedure for checking access authorization distinguishes between read access (send file) and write access (receive file). If the user does not have the appropriate access authorization, the transfer request is rejected.

A distinction must be made between the following cases:

- RACF is installed and active:

  openFT uses RACF to check the user's authorization to access the send or receive file (read or write access). The RACROUTE macro makes use of the RACF macro RACHECK with the resource class DATASET for this purpose. For technical reasons the RACROUTE macro again calls on the RACF macro RACINIT to supply the user ID specified in TRANSFER-ADMISSION, generally together with the associated user password.

- RACF is not installed or is not active:

  In this case, the user's authorization to access a file is checked only in the case of a receive file which is password-protected according to the catalog entry. The file password specified in the NCOPY command for the receive file is then checked against the relevant entry in the PASSWORD file of the system (PROTECT macro). No password check takes place for send files (which can only be read).

2.7.3 Checking authorization for follow-up processing

Users must demonstrate their authorization to initiate follow-up processing by specifying a PROCESSING-ADMISSION. If the PROCESSING-ADMISSION is not explicitly specified, the data are taken from the TRANSFER-ADMISSION for the PROCESSING-ADMISSION. In the case of follow-up processing, the parameters USER-ID, ACCOUNT and PASSWORD must be explicitly assigned a value in one of the two ADMISSIONs. When FTAC is used, the data for the PROCESSING-ADMISSION can also be specified in an admission profile.

The authorization to initiate follow-up processing is checked by openFT.

The description given in "section “Checking the transfer admission” on page 91" applies when the authorization to initiate follow-up processing is checked by openFT.

A special case of follow-up processing under openFT is the character string "ALLOC DSNAME(...)". This special openFT statement is used to specify the name of a cataloged PS data set or the name of a member of a cataloged PO or PDSE data set containing a complete executable job.

openFT starts this job within follow-up processing via the Internal Reader. In this case, openFT does **not** generate any additional job control statements. In this way, it is possible to execute follow-up processing jobs with user-specific job parameters.

In this special case:

- openFT checks the access authorization of the user to this file on the basis of the data in the PROCESSING-ADMISSION. This transfer admission check is always performed after successful checking of the authorization for initiating follow-up processing.
- However, openFT does **not** check the values for user ID, account number and password specified in this PS data set or in this member. In order to prevent users from accessing the system unchecked, you are therefore recommended to use openFT-AC.

2.7.4 Checking preprocessing and postprocessing authorizations

The authorization to perform preprocessing and postprocessing corresponds to the admission under which the file transfer itself was performed (see "Checking access authorization"). It is therefore not taken from the PROCESSING-ADMISSION. The admission for z/OS is either proven explicitly on the basis of the USER-ID, ACCOUNT and PASSWORD specifications in the local TRANSFER-ADMISSION or implicitly through the use of an admission profile. In an admission profile that is to be used for preprocessing or postprocessing, the USER-ID, ACCOUNT and PASSWORD specifications must be stored in the USER-ADMISSION.

2.8 Configuring FTAC

Authorization of the FTAC administrator

It is recommended that the authorization to administer FTAC be given to those users in the system who are responsible for data protection in an z/OS system, since they are the best placed to know what protection measures are required where.

The FTAC administrators of an openFT instance are defined in the FTACADM member of the PARM parameter library (see page 62).

Adapting the default admission set

After the installation of FTAC, all values of the default admission set are **set at 0**!

This means that it is not yet possible to execute a file transfer with the local openFT instance. This is because as long as no other admission sets are made with FTMODADS, the default admission set is valid for all user IDs. The maximum security level 0 for the basic functions means that these basic functions may not be used. An FTAC administrator must therefore use the command FTMODADS to raise the values of the default admission set.

Examples

1. All partner systems should be accessible for file transfer for all FTAC users. This is achieved by setting all the values of the default admission set to100. The following command is used:

   ```
   FTMODADS␣*STD,MAX-LEV=100
   ```

   More information on the command FTMODADS can be found starting on page 248.

2. A differentiated setting of the default admission set might look as follows:

   ```
   FTMODADS USER-IDENTIFICATION=*STD,                                -
             MAX-LEVELS=(OUTBOUND-SEND=50,OUTBOUND-RECEIVE=50,       -
                         INBOUND-SEND=20,INBOUND-RECEIVE=20,         -
                         INBOUND-PROCESSING=10,INBOUND-MANAGEMENT=0)
   ```

   The different security levels are assigned selectively. For example, the function "inbound management" can be fully blocked by setting the security level to 0.

WARNING!
Note that FTAC is only effective for connected products such as openFT. If other file transfer products without an openFT-AC connection are also being used, a more comprehensive and coordinated security concept would be advisable.

A key pair set must be created with FTCREKEY to be able to use the Crypto module.

3 Operation of openFT

This chapter contains information on the subject of administration, security and control and monitoring functions.

FT and FTAC administration

Whereas FT users can only monitor and manage their own FT requests; the FT administrators are able to access all FT activities taking place on their system.

FTAC administration is independent of FT administration. The FTAC administrators are the security managers for FT activities on your computer. They have ultimate authority concerning all admission sets and profiles.

As of openFT V11, it is also possible to set up a remote administration server and use this to administer several openFT instances from any client on a z/OS system. See the chapter “Central administration” on page 155.

The FT administrator

Before you can administer an openFT instance, your user ID must be entered in the FTADM member of this instance’s PARM parameter library. You have the following options:

- You can administer openFT using simple TSO commands. To do this, you must work under a TSO user ID which is authorized to access the library containing these TSO commands (FT administrator ID). The user IDs which have this authorization are specified during openFT installation (see section “User IDs for openFT” on page 31).
- Alternatively, you can also use a convenient ISPF menu interface to administer openFT. The requirements are the same as for administration using TSO commands. In addition, your TSO user ID must be authorized to access libraries containing the menu interface members (such as the ISPF panel definition, for example).
- openFT may also be controlled from an operator console. In addition, you can use network management systems such as NetView® or compatible products for controlling openFT.

Tasks of the FT administrator

As the FT administrator, you are responsible for the administration and smooth operation of an openFT instance. For this reason, you are authorized to perform special tasks. You can

- manage the request file and the partner list,
- activate and deactivate the openFT instance,
- control the resources used,
- administer key pair sets for encryption and authentication,
- request information about the openFT instance,
- monitor the openFT instance,
- request information about the FT requests of all users,
- cancel/abort the file transfer requests of all users,
- deactivate and reactivate FT requests submitted in the local system to individual remote FT systems,

These FT administrator tasks are described in detail in the following sections.

The FTAC administrator

When using the FTAC functionality (separate delivery unit openFT-AC), one or more of the user IDs will be authorized to administer and control the FTAC functions. FTAC administrators are defined in the FTACADM member of an openFT instance's PARM parameter library. The tasks of the FTAC administrator are described in section "Administrating and controlling FTAC functions" on page 126.

3.1 Optimizing the operating parameters

The proposals listed below suggest a number of ways in which the FT administrator can optimize FT operation by modifying the operating parameters. The command used for this purpose is FTMODOPT. It is always advisable to alter only one operating parameter at a time, so that the precise effects of the change can be observed.

Problem	**Suggested solution**
Poor dialog response times	1. Reduce TRANSPORT-UNIT-SIZE 2. Reduce CONNECTION-LIMIT
Computer overloaded, network load not yet optimized	1. Set PROC-LIMIT to 1 or 2 2. Increase TRANSPORT-UNIT-SIZE 3. Reduce CONNECTION-LIMIT
Computer and network overloaded	1. Set PROC-LIMIT to 1 or 2 2. Reduce CONNECTION-LIMIT
Throughput inadequate	1. Increase TRANSPORT-UNIT-SIZE 2. Under TCP/IP: Set RFC1006 transport protocol (see page 195: FTADDPTN)
Prolonged requests block other requests	1. Increase CONNECTION-LIMIT
Requests to a particular partner system use up all resources	1. Increase CONNECTION-LIMIT
Some Requests are present in the request file for a very long period (several days or weeks) without being processed.	1. Set MAX-REQUEST-LIFETIME

3.1.1 Interdependencies for optimized parameterization

The optimum settings for operating parameters depend on several different constraints:

- load levels of the local and remote systems,
- load level in the network,
- line transfer rates in the network,
- network structure (connection paths reserved for FT or shared paths for FT and dialog operation),
- incorporation of gateway computers
- type, performance or generation of the transport system used,
- average size of files to be transferred,
- number of files to be transferred (e.g. per day).

In some instances, these boundary conditions are themselves subject to dynamic change (load levels for example), so it is not possible to calculate in advance the optimized values for a particular installation.

3.1.2 Achieving optimized operation

Experience has shown that the most suitable parameter settings can only be achieved in stages.

Initially the openFT default values should be left unchanged. In most cases it will be possible to run file transfers satisfactorily using these parameter values.

If not, however, as a second step an improvement can be sought by changing **one** of the parameter values. It is normally not advisable to change more than one parameter at a time as otherwise there is no way of ascertaining the precise effect of each change.

If satisfactory operation of the FT system has still not been achieved, the FT administrator can repeat the second step, changing a different parameter.

The FT administrator can control the operation of the FT system using the parameters PROCESS-LIMIT, CONNECTION-LIMIT, TRANSPORT-UNIT-SIZE and MAX-REQUEST-LIFETIME. These parameters are discussed in the sections below. In addition, the effect of changing the parameters is also described.

3.1.3 Changing the PROCESS-LIMIT operating parameter

The PROCESS-LIMIT parameter defines the maximum number of tasks that may be used for processing file transfer requests. The number of file transfer requests per task handled simultaneously can be expressed as follows:

$$\frac{\text{CONNECTION-LIMIT}}{\text{PROCESS-LIMIT}}$$

CONNECTION-LIMIT is the maximum number of parallel transport connections that can be used to execute requests.

If the PROCESS-LIMIT value remains fixed and the value of CONNECTION-LIMIT is increased, then proportionately more transport connections are available for each task and therefore more requests can be processed per task. The reduction of the PROCESS-LIMIT value where CONNECTION-LIMIT remains constant achieves the same effect. If the value of the quotient is reduced (by reducing CONNECTION-LIMIT or increasing PROCESS-LIMIT), a smaller proportion of transport links is available per task. Consequently, fewer requests can be processed per task.

If the number of requests awaiting processing exceeds the value of the quotient but the number of tasks assigned has not reached the PROCESS-LIMIT value, then another task is initiated.

Higher PROCESS-LIMIT:

- fewer wait times for input/output
- better use of potentially underutilized computer resources

Lower PROCESS-LIMIT:

- reduced load on the local system

3.1.4 Changing the CONNECTION-LIMIT operating parameter

The CONNECTION-LIMIT parameter defines the maximum number of transport connections to be used in the execution of file transfer requests. Since the processing of a request always requires a new transport connection to be set up, CONNECTION-LIMIT also defines the maximum number of requests the system can process in parallel.

A third of the connections is reserved for outbound requests and a third for inbound requests. The remaining third can be used for inbound or outbound requests as required.

In order to obtain the same level of throughput with your openFT partners, it may therefore be necessary to increase the CONNECTION-LIMIT value.

Higher CONNECTION-LIMIT:

- increased data throughput
- better use of potentially underutilized processor capacity.

Lower CONNECTION-LIMIT:

- reduced load on the local system and network, and hence less or even no impact upon interactive operation.

3.1.5 Changing the TRANSPORT-UNIT-SIZE operating parameter

The TRANSPORT-UNIT-SIZE parameter defines the maximum length of the message transmitted to the transport system by openFT. TRANSPORT-UNIT-SIZE has no effect for links to FTAM partners. Message flow control ensures that only a specific number of messages are being transmitted across the network at any one time. The TRANSPORT-UNIT-SIZE parameter enables the administrator to control the amount of FT data present in the network at a particular time. The value specified for TRANSPORT-UNIT-SIZE can be changed by the remote system or by the transport system (maximum message length).
A maximum value of 32767 is recommended for TRANSPORT-UNIT-SIZE. This value is the default value when creating a new request file.

Higher TRANSPORT-UNIT-SIZE:

- increased data throughput
- reduced load on the local system since fewer calls to the transport system are necessary.

Lower TRANSPORT-UNIT-SIZE:

- reduced load on the network
- the time required to transmit an FT message across a communication link is reduced, which in turn decreases the wait time for messages from other users. For slow communication links, response times can, for example, be improved in interactive mode.

3.1.6 Setting the MAX-REQUEST-LIFETIME operating parameter

The MAX-REQUEST-LIFETIME parameter is used to set a global limitation for the lifetime of openFT requests. Whereas it was previously necessary to use NCANCEL to remove requests without a lifetime limitation from the request file, you can now specify a maximum lifetime (calculated from the start time). The maximum lifetime applies to both inbound and outbound requests and is specified in days.

3.2 Administering code tables

A code table is a table that describes a character set (Coded Character Set, CCS). It describes a set of characters and the way in which they are coded (see the example on page 106). Different systems frequently use different character sets:

- Internally, Unix systems and Windows systems use an ISO-8859-x code described in ISO 8859. ASCII (American Standard Code for Information Interchange) is a subset of ISO-8859-1. Character set CP1252, which is often used on European Windows systems is a superset of the ISO-8859-1 character set.
- BS2000/OSD systems, OS/400 or z/OS computers, on the other hand, generally use a variant of EBCDIC (Extended Binary-Coded Decimal Interchange Code) internally.

Different character sets are of significance when transferring text files, but not when transferring other file formats (binary, transparent, etc.), as openFT does not convert the contents of the file in this event.

In the case of partner systems up to openFT V9, all characters from the ISO-8859-x character sets are supported. With partner systems as of openFT V10, the complete unicode character set is supported. See also the table on page 104. When transferring text files, openFT converts the contents if this is necessary. Here, it is important that the character sets in the communicating systems are compatible, i.e. the characters used on the send side must also be available in the character set on the receive side.

CCS name

Each character set is identified by a CCS name, which means that the character set for a file can be specified by a CCS name (e.g. ISO88591, EDF041, UTF8, IBM037). This can be done in three ways:

- by the file transfer request itself. This specification takes priority.
- by the assignment table between CCS names and file name patterns (FNAMECTB member in the openFT parameter library, see page 80). This allows character sets to be assigned on a file-specific basis. If FTAC is used, this file name can be made up of the specification in the transfer request and specifications in the admission profile accessed during the transfer request.
- by the default setting in the operational parameters (FTMODOPT command, CODED-CHARACTER-SET=, default IBM1047). This setting applies if no assignment was made in the transfer request or in the FT parameter library.

A range of character sets is already integrated in openFT (see page 104), but there are a number of IBM-specific variants of EBCDIC that represent special characters such as ä, ö, [, { in a different way from the common EBCDIC DF04 codes. For this reason, the FT administrator must be able to set up additional code tables in z/OS systems, containing special character sets and to which a CCS name is assigned. How to create your own code table is described as of page 105.

Available character sets and code tables supplied

The following character sets are completely integrated in openFT:

Name of the CCS	Meaning
ISO88591 to ISO8859B and ISO8859D to ISO8859G	for the ASCII tables ISO8859-1 to ISO8859-11 and ISO8859-13 to ISO8859-16
ISO646	for the international 7-Bit ASCII table
ISO646DE	for the german 7-Bit ASCII reference version
EDF041 to EDF04A and EDF04D to EDF04F	for the EBCDIC tables DF04-1 to DF04-10 and DF04-13 and DF04-15
EDF03IRV	for the international 7-Bit EBCDIC table
EDF03DRV	for the german 7-Bit EBCDIC table
UTF16	for Unicode with UTF-16 coding (platform-specific endian)
UTF8	for Unicode with UTF-8 coding
UTFE	for Unicode with the UTF-E coding
UTF16LE	for Unicode with UTF-16 coding (little-endian)
UTF16BE	for Unicode with UTF-16 coding (big-endian)
UTFEIBM	for Unicode with the UTF-EBCDIC coding defined by IBM
CP1252	for ANSI character set with Euro symbol defined by Microsoft (s.o.)
IBM1047	for the OpenExtensions EBCDIC character set defined by IBM
CP850	for the OEM character set defined by Microsoft

In addition, the code tables IBM037, IBM273 and IBM500 are stored in <openFT installation directory>.OPENFT.SYSCCS on installation. These tables were previously named FTCP037, FTCP273 and FTCP500.

Creating code tables for custom character sets

When migrating from openFT Version V9 (or earlier) to V10, custom code tables must be converted to the new format. It should be noted that the code tables in V9 converted in both directions between EBCDIC.DF.04-1 and the relevant custom code. In V10, the custom code must be mapped to UTF-16, which corresponds to mapping to ISO8859-1 (each character being prefixed by a 00 byte). Newly created code tables now allow characters to be represented that are not contained in ISO8859-1.

You must save the code tables as members in <openft qualifier>.<inst>.SYSCCS. This PO library is empty after installation.

The member name is the CCS name of the associated character set. Assembly is no longer required.

Structure of a code-conversion table

The text file must have the following structure:

- The first line starts with a '#'.

  The second character is an blank. The remainder of the line contains a comment which characterizes the code contained.

- The second line contains an alphabetic character which can at present only have the value 'S'. 'S' stands for single-byte code, i.e. a character is always 1 byte in length.

- The third line contains three numbers.

  The first number is a 4-digit hexadecimal number. This defines the substitution character to be used if a Unicode character cannot be mapped to the code.

  The second number is currently always '0'.

  The third number is a decimal number which defines the number of code pages that follow. It currently always has the value '1'.

- The following lines define the code pages and have the following structure:

  – The first of these lines contains the number of the code page in the form of a two-digit hexadecimal number. Currently, only code page 00 is permitted.

  – Each of the subsequent lines contains the assignment of a character to the corresponding 8-bit code position. A character is represented by its UTF-16 code in the form of a four-digit hexadecimal number. The values are arranged in 16 lines, each of which contains 16 4-digit hexadecimal numbers with no spaces.

Example for ISO8859-15 (Western Europe with Euro symbol)

```
# Encoding file: iso8859-15, single-byte
S
003F 0 1
00
0000000100020003000400050006000700080009000A000B000C000D000E000F
0010001100120013001400150016001700180019001A001B001C001D001E001F
0020002100220023002400250026002700280029002A002B002C002D002E002F
0030003100320033003400350036003700380039003A003B003C003D003E003F
0040004100420043004400450046004700480049004A004B004C004D004E004F
0050005100520053005400550056005700580059005A005B005C005D005E005F
0060006100620063006400650066006700680069006A006B006C006D006E006F
0070007100720073007400750076007700780079007A007B007C007D007E007F
0080008100820083008400850086008700880089008A008B008C008D008E008F
0090009100920093009400950096009700980099009A009B009C009D009E009F
00A000A100A200A320AC00A5016000A7016100A900AA00AB00AC00AD00AE00AF
00B000B100B200B3017D00B500B600B7017E00B900BA00BB01520153017800BF
00C000C100C200C300C400C500C600C700C800C900CA00CB00CC00CD00CE00CF
00D000D100D200D300D400D500D600D700D800D900DA00DB00DC00DD00DE00DF
00E000E100E200E300E400E500E600E700E800E900EA00EB00EC00ED00EE00EF
00F000F100F200F300F400F500F600F700F800F900FA00FB00FC00FD00FE00FF
```

3.3 Administering requests

You can use the NSTATUS command (see page 371ff) to view information on selected FT requests. Possible selection criteria include

- the user ID,
- the system which initiated the request,
- certain statuses of FT requests, and
- names of file or job variables affected by an FT request in the local system.

The FTMODREQ command permits both administrator and user to modify the order and priority of outbound requests of openFT and FTAM partners within the request queue.

The NCANCEL command enables you to remote FT requests from the request queue or to abort file transfer while in progress. The selection criteria at your disposal are much the same as those for the NSTATUS command. .

FTMODPTN allows you to activate or deactivate locally submitted requests for a particular remote system (see STATE, page 296).

3.4 Administering partners

openFT offers the FT administrator the following commands for the administration of partner systems:

FTADDPTN	Add new partner system entries to the partner list
FTMODPTN	Modify partner system entries in the partner list
FTREMPTN	Remove partner systems from the partner list
FTSHWPTN	View information on partner systems in the partner list and save the partner list as a command procedure
FTMODOPT	Enable/disable dynamic partners

3.4.1 Defining partner properties

You use the FTADDPTN command to define the properties of partners:

- Partner address, see page 108
- FTAC security levels, see page 111
- Automatic deactivation, see page 112
- Partner-specific trace settings, see page 139
- Authentication setting and instance identification for the partner, see page 113
- Sender verification, see page 118
- Priority, only takes effect if the request priority is the same. See page 199.

You can modify these settings whenever you want with FTMODPTN.

3.4.1.1 Specifying partner addresses

The addressing of partner systems has been modified in openFT V10 in terms of both syntax and function scope:

- the structure of the partner address has been considerably simplified and adapted to comply with internet address conventions, see “Structure of the partner address”. You specify the partner address as in the past in the FTADDPTN or FTMODPTN command.
- a partner can be accessed directly via its address in FT requests even if it is not entered in the partner list. This is only possible if the “dynamic partner” function is enabled, see page 112. The partner entry ANYBODY is not used in this case.
- It is now also possible to address FTP partners.

Structure of the partner address

A partner address has the following structure:

[protocol://]host[:[port].[tsel].[ssel].[psel]]

host (= computer name or processor name, see page 109) is mandatory; all other specifications are optional. In many cases, the other specifications are covered by the default values, so that the host name suffices as the partner address, see “Examples” on page 110. Final ’.’ or ’:’ can be omitted.

The individual components of the address have the following meanings: The individual components of the address have the following meanings:

protocol://

Protocol stack via which the partner is addressed. Possible values for *protocol* (uppercase and lowercase are not distinguished):

openft	openFT partner, i.e. communication takes place over the openFT protocol.
ftp	FTP partner, i.e. communication takes place over the FTP protocol.
ftadm	ADM partner, i.e. communication takes place over the FTADM protocol for remote administration and ADM traps.

Default value: **openft**

host

Computer name via which the partner is addressed. Possible entries:

- internet host name (e.g. DNS name), length 1 to 80 characters, up to 24 characters for z/OS partner systems
- TNS name from the z/OS library (TNSTCPIP member), up to 8 characters in length.
- SNA LU name, length 1 to 8 characters
- IPv4 address with the prefix %ip, i.e. for example %ip139.22.33.44
  The IP address must always be specified as a sequence of decimal numbers separated by dots and without leading zeros.

port

When a connection is established over TCP/IP, you can specify the port name under which the file transfer application can be accessed in the partner system.
Permitted values: 1 to 65535;

In the case of an SNA-LU connection, (*host* = LU name) you must specify the value *sna* for the port number.

Default value: **1100** for openFT partners
A different default value can also be set in the operating parameters using FTMODOPT

21 for FTP partners

11000 for ADM partners

tsel

Transport selector under which the file transfer application is available in the partner system. The transport selector is only relevant for openFT and FTAM partners.
You can specify the selector in printable or hexadecimal format (0xnnnn...):

Length, 1 through 8 characters; alphanumeric characters and the special characters # @ $ are permitted. A printable selector will be coded in EBCDIC in the protocol and may be padded with spaces internally to the length of eight characters.

Default value: **$FJAM**

ssel

Session selector under which the file transfer application is accessible in the partner system. You can specify the selector in printable or hexadecimal format (0xnnnn...).

Length, 1 through 10 characters; alphanumeric characters and the special characters @ $ # _ - + = * are permitted. A printable selector will be coded as variable length ASCII in the protocol.

Default value: empty

psel

Only relevant for FTAM partners, not used under z/OS.

Examples

The partner computer with the host name FILESERV is to be addressed over different protocols/connection types:

Connection type/protocol	**Address specification**
openFT partner	FILESERV
FTP partner	ftp://FILESERV
SNA partner via openFT protocol (FILESERV is the LU name)	FILESERV:sna

3.4.1.2 FTAC security levels for partner entries

If the FTAC functionality is to be used, the FT administrator must additionally define the appropriate FTAC security level for each partner entry using the command FTMODPTN (operand SECURITY-LEVEL). This must be done in cooperation with the FTAC administrator.

The security levels regulate the degree of protection with respect to the partner system. This protection can be best determined by the FTAC administrator. Therefore, he should advise the FT administrator on the assignment of the security levels to the partner systems. A high security level is used when a high degree of security is required, and a low level for a low degree of security. When FTAC is first installed, the security levels should be assigned in multiples of ten. This leaves the option open to incorporate new partner systems flexibly into the existing hierarchy.

If the degree of required security changes with respect to a partner system, the security level of the partner system can be modified with the command FTMODPTN to meet the new requirements.

You can also use the operand SECURITY-LEVEL=*BY-PARTNER-ATTRIBUTES to activate the following automatic mechanisms for the security levels:

- Partners that are authenticated by openFT are assigned security level 10.
- Partners that are known in z/OS (i.e. they are addressed via their VTAM or DNS name for example) are assigned security level 90.
- Partners which are accessed only via their IP address (e.g. FTP partners) are assigned security level 100.

This automatic mechanism can be activated on a partner-specific basis (FTADDPTN and FTMODPTN) or globally by means of FTMODOPT.

If the security level is not specified at the partner system or if the partner system is not defined in the partner list then openFT uses the global settings in the operating parameters (FTMODOPT). Here, it is also possible to specify a fixed security level as the default.

The security level of a partner entry is taken into account when a user wants to process a request via this partner entry. FTAC compares the security level of the partner entry with the security level for this function (e.g. inbound sending) specified in the user's admission set. If the security level in the admission set is lower than that in the partner entry, the request is rejected by FTAC. If a privileged FTAC profile is used for the request, it can override the restrictions defined in the admission set.

3.4.1.3 Automatic deactivation

The FTMODPTN command can be used to assign an “automatic deactivation” attribute to openFT partner systems. Automatic deactivation disconnects the partner after five failed attempts to establish a link. This prevents unnecessary costs from arising in the case of certain link types, which also charge for unsuccessful link establishment attempts. Before any new attempts are made, the system must be manually reactivated.

3.4.2 Dynamic partners

Users may, as required, specify partners via their names in the partner list or via their addresses. In this way, they can also address partners that are not entered in the partner list.

Dynamic partners are partners which are either not entered in the partner list or which are entered in the partner list without a name (PARTNER-NAME=*NONE in the FTADDPTN command). Hence, a user can specify a dynamic partner only via its address.

If a partner is not entered in the partner list then the same automatic mechanism applies for the security level as described for SECURITY-LEVEL=*BY-PARTNER-ATTRIBUTES, see page 111.

As FT administrator, you may also lock the use of dynamic partners for security reasons. You do this by means of the FTMODOPT command with the operand DYNAMIC-PARTNERS=*OFF. In this case, it is necessary to address partners via their names in the partner list. They cannot be addressed directly via their address. Inbound access is then also only permitted to partners that are entered in the partner list.

3.4.3 Backing up the partner list

You can back up the entries in the partner list by means of the FTSHWPTN command. FTSHWPTN outputs the partner entries in the form of FTMODPTN commands. To do this, specify the OUTPUT(LAYOUT=...) operand.

3.5 Security in FT operation

A user wanting to access resources of a system must always provide the system with proof of his or her authorization for the access. In the case of file transfer activities, access admission must be verified in both the local and the remote system. Verification usually entails specifying a user ID and a corresponding password.

An even higher level of security in file transfer is offered FTAC. FTAC provides the functions for controlling FT activities on a computer-specific and user-specific basis. The FTAC administrator defines admission profiles for each user ID and if FTAC is in use, the FT administrator must assign security levels to the partner systems (see FTADDPTN and FTMODPTN SECURITY-LEVEL= and section “FTAC security levels for partner entries” on page 111).

⚠ **Warning!**
Note that openFT-AC is only effective for connected products such as openFT. If other file transfer products without an openFT-AC connection are also being used, a more comprehensive and coordinated security concept would be advisable.

Notes on Secure FTP

A standard Secure FTP server makes its key and the certificate available to the openFT outbound client for encryption purposes. No mutual authentication is carried out.

An openFT client is able to exchange encrypted outbound user data with a standard Secure FTP server if openFT-Crypt is installed on the openFT side and the FTP server supports the TLS protocol. AES is used as the encryption method.

If the openFT client requires encryption of the user data in the request, but the FTP server does not support the TLS protocol, the request is rejected. If the openFT client does not require encryption of the user data, the login data is only encrypted if the FTP server accepts the TLS protocol, otherwise the login data is transferred in unencrypted form.

3.5.1 Authentication

If data requiring a high degree of security is to be transferred, it is important to subject the respective partner system to a reliable identity check (“authentication”). The two openFT instances taking part in a transfer must be able to cryptographically check one another to determine whether they are connected to the “correct” partner instance.

Therefore, as of versions openFT V8.1 for Unix system and Windows systems, and V9.0 for BS2000 and z/OS, an expanded addressing and authentication concept is supported for openFT partners. It is based on the identification of openFT instances using a network-wide, unique ID and exchanging partner-specific key information.

You should note that authentication in openFT for z/OS is only possible for partners that are entered in the partner list!

When communicating with partners that are using openFT version 8.0 (or earlier), the functions described in the following are not usable. The identity can be detected via authentication check as before.

Instance identification

Each openFT instance that works with authentication must be assigned a network-wide, unique instance ID. This ID replaces the previous identification of openFT instances using processor and application names. The instance ID is a unique name, up to 64 characters long. Its uniqueness must be based on something other than case-sensitivity. An instance ID may be comprised of alphanumeric characters and the special characters. You are advised only to use the special characters ".", "-", ":" or "%". The first character must be alphanumeric or the special character "%". The "%" character may only be used as a first character. An alphanumeric character must follow a "." character.

In order to ensure the network-wide uniqueness of instance IDs, you should proceed as follows when assigning them:

– If the openFT instance has a network address with a **DNS name**, you should use this as the ID. You can create an "artificial" DNS name for an openFT instance, by placing part of a name, separated by a period, in front of an existing "neighboring" DNS name.

– If the openFT instance does not have a DNS name, but is connected to a TCP/IP network, you should use the ID **%ipn.n.n.n** (n.n.n.n is the IP address of the local openFT instance, minus the leading zeros in the address components).

You allocate these IDs for your local openFT instances using the IDENTIFICATION parameter of the FTMODOPT command.

The form of instance ID used internally by openFT for partners using a version earlier than V8.1, (i.e. **%.<processor>.<entity>)**, should not be used for your own openFT instance.

Store instance IDs of partner systems in the partner list using the IDENTIFICATION parameter of the FTADDPTN command, or FTMODPTN. With the aid of the partner systems' instance IDs, openFT manages the resources assigned to those partners, such as request hold queues and cryptographic keys.

Creating and managing local keys

In order to ensure that your own openFT instance can be authenticated in the partner system, a suitable public key for the instance must be made available to the partner system. Using CREATE-FT-KEY-SET, you can create an RSA key pair set for the local openFT instance; this consists of a private (secret) key and a public key. Private keys are internally administered by openFT. Public keys are stored under the OPENFT QUALIFIER of the openFT instance under the following name:

```
<inst>.SYSPKF.R<key reference>.L<key length>
```

The key reference is a numeric designator for the version of the key pair, the default key length is currently 768 in z/OS. The public key files are text files, which are created in the character code of the respective operating system, i.e. EBCDIC.DF04-1 for BS2000 and z/OS, ISO8859-1 for Unix and Windows systems. If the file is transferred as a text file it is automatically converted in accordance with the available code conversion tables.

In a **<inst>.SYSPKF.COMMENT** file on the OPENFT QUALIFIER of the openFT instance, you can store comments, which are written in the first lines of the public key files when a key pair set is created. Comments could, for example, contain the contact data for the FT administrator on duty, the computer name, or similar information that is important for partners. The lines in the SYSPKF.COMMENT file may be a maximum of 78 characters in length. Using the FTUPDKEY command, you can import updated comments from this file into existing public key files at a later time.

If a public key file has been unintentionally deleted or otherwise manipulated, you can re-create the public key files of the existing key pair sets using FTUPDKEY.

If you want to replace a key pair set with a completely new one, you can create a new key pair set using FTCREKEY. You can identify the most current public keys by the highest value key reference in the file name. *Open*FT supports a maximum of three key pair sets at a time. The existence of several keys should only be temporary, until you have made the most current public keys available to all the partner systems. Afterwards, you can delete the key pair sets no longer needed using FTDELKEY.

It must be ensured that each openFT administrator being responsible for the keys has access to the SYSPKF files and the <inst>.SYSKEY library on the OPENFT QUALIFIER of the openFT instance. This can be done, either by assigning operating system-specific access rights or by setting up corresponding FTAC admissions profiles.

i In the case of partners using openFT as of V11.0, a 256-bit AES key is used and in the case of partners using openFT up to V10.0, a 128-bit AES key is used.

Distributing the keys to partner systems

Distributing the public key files to your partner systems should take place by secure means, for example by

- distribution by cryptographically secure e-mail
- distribution on a CD (by courier or by registered mail)
- distribution via a central openFT file server, the public keys of which are in the partners' possession

If you transmit your public key files to partners using Unix or Windows systems, you must ensure that these files are re-coded from EBCDIC to ISO 8859-1 (e.g. by transferring them as a text file via openFT).

The public key file of your local openFT instance is stored in the partner system in the following location:

- For partners with openFT for BS2000, as a type D PLAM element in the **SYSKEY** library, the configuration user ID of the partner instance. The partner name allocated for your openFT instance in the remote network description file or partner list must be selected as the element name.
- For partners with openFT for Unix systems, in the **/var/openFT/<instance>/syskey** directory. The instance ID of your local openFT instance must be selected as the file name. The file name must not contain any uppercase letters. If the instance ID contains uppercase letters, these must be converted to lowercase in the file name.
- For partners with openFT for Windows, in the **openFT\var\<Instance>\syskey** directory. The instance ID of your local openFT instance must be selected as the file name.
- For partners with openFT for z/OS, as a PO element in the **<inst>.SYSKEY** library. The partner name allocated for your openFT instance in the remote network description file or partner list must be selected as the element name.

Managing the keys of partner systems

The public keys of the partner systems are stored in BS2000 as type D PLAM elements in the **<inst>.SYSKEY** library under the OPENFT QUALIFIER of the local openFT instance. The partner name of the partner system as defined in the partner list must be selected as the element name. If an updated public key is made available by the partner instance, the old key must be overwritten by it.

While the SYSKEY library is open for updating, openFT is unable to perform any authentication of inbound requests and new requests are rejected. You should therefore make sure that the library is not open for long, for example by entering the updated members in

SYSKEY via openFT. If you stop openFT to work on SYSKEY (with FTSTOP) then new restartable inbound requests are stored in the partner systems and are subsequently processed automatically.

Local and partner authentication

There are three distinct usages:

- The local openFT instance checks the identity of the partner instance. This assumes that a current, public key of the partner instance was stored locally in the SYSKEY library, the name of which corresponds to the partner name of the partner instance defined in the partner list. This sort of configuration makes sense, for example, if files on a file server are to be accessed using openFT. It is important for the local openFT instance, that the retrieved data should come from a reliable source (from the authenticated partner). In contrast, the file server is not concerned with who is accessing it.
- The partner instance checks the identity of the local openFT instance. This assumes that a current, public key of the local openFT instance is stored in the partner instance (re-coded - for Unix and Windows partners). This sort of configuration would be considered, for example, if partner systems in several branch systems are to be accessed using openFT from a central computer, and where the branch system computers only allow the central computer access (and, in practice, only the central computer).
- Both of the openFT instances engaged in a transfer authenticate each other. This assumes that current, public keys were mutually exchanged and the partners are addressing each other using their instance IDs. In this way, it can be ensured that the data not only comes from a reliable source, but that it will also end up in reliable hands.

3.5.2 Extended authentication check

openFT partners using openFT from version 8.1 onwards, support the authentication mechanism (see page 113). If the local system has a public key of the partner at its disposal, the partner's identity is checked by cryptographic means.

For partner systems that do not work with authentication, inbound requests are checked with the aid of the processor name, in order to ascertain whether the calling system has a valid entry in the partner list. As an option, openFT offers the possibility of checking (via sender checking) not only the processor name, but also the transport address.

The extended sender checking can be globally enabled for openFT partners or just for specific partners:

- globally, using
  FTMODOPT... PARTNER-CHECK=*TRANSPORT-ADDRESS
- only for specific partners, using
  FTADDPTN ... PARTNER-CHECK=*TRANSPORT-ADDRESS or
  FTMODPTN ... PARTNER-CHECK=*TRANSPORT-ADDRESS

The global setting is valid for all partners with the value
PARTNER-CHECK=*BY-FT-OPTIONS (default in the FTADDPTN).

Use of the extended authentication check while dynamic partners are also enabled is only of value if FTAC functionality is being used. Even if the extended authentication check is enabled, it will not be evaluated if dynamic partners are also enabled.

If the authentication check returns a negative result, the request is rejected.

3.5.3 Protection mechanisms for file transfer

openFT supports for openFT partners the encryption of the data sent and received in the process of setting up the connection and processing a file transfer request. The partners involved in file transfer automatically negotiate encryption and use of the appropriate public key in the process of connection set-up. openFT automatically encrypts the request description data provided that the partner supports this functionality.

Using the FTCREKEY command, the FT administrator must create at least one key pair set, upon which the encryption will be based and carried out.

If the file content is to be encrypted for transfer by openFT, the optional openFT-CR component must be installed.

The encrypted transfer of file contents is requested with the NCOPY command and only executed if openFT is installed with openFT-CR in both the systems involved in the file transfer. If one of the two systems is not capable of handling encrypted file transfers, the request is rejected with the message FTR2111 (no openFT-CR in local system) or with FTR0055 (encryption is not possible in remote system).

For legal reasons, openFT-CR is not available in all countries.

3.5.4 Protection mechanisms against file manipulation and corruption

During communications with openFT partners, openFT as of V8.1 implicitly checks the integrity of the transferred data. For requests with unencrypted file content, the integrity of the request description data is checked. For requests with encryption, the integrity of the transferred file content is also checked. If an error is detected, restartable requests attempt a new transfer. Non-restartable requests are aborted.

In this way it is possible to detect and prevent malicious manipulations of the transferred data (e.g. in insecure public networks such as the Internet).

Errors on the physical transfer channels are identified and rectified by the communication system itself. No data integrity check at openFT level is required for this.

3.6 Monitoring and controlling FT operation

Fetch information on the FT system

The FT administrator uses the following commands to obtain information on the system:

FTSHWOPT	Information on operating parameters
FTSHWPTN	Information on partner systems
FTSHWLOG	Information on log entries
NSTATUS	Information on file transfer status
FTSHWINS	Information on openFT instances
FTSHWMON	Show monitoring data from openFT operation

The FTSHWOPT command furnishes information on the current settings of the operating parameters.

FTSHWPTN yields information on the partner systems and their associated properties, e.g., names, addresses, security levels for FTAC, and so on. The command and the possible outputs are described in detail starting on page 352.

To support automatic monitoring, some events which are not direct responses to user input are reported by openFT via console messages. More detailed information on this topic can be found in the section "FT logging" on page 121.

The command FTSHWLOG can be used to display the logs of file transfer requests. You will find more information on this subject in the section below and in the description of the FTSHWLOG command on page 311ff.

NSTATUS enables the FT administrator to retrieve information on all file transfer requests in his or her system, even when the FT system is stopped.

Using FTSHWINS, the FT administrator can find out which openFT instances exist in the system and have their characteristics and status displayed. FTSHWINS only works if openFT has been started as a subsystem.

FTSHWMON outputs the monitoring values from openFT operation. To do this, monitoring must be activated by means of FTMODOPT.

3.6.1 FT logging

The following 3 commands are available for the FT logging function:

FTDELLOG	Delete logging records
FTMODOPT	Switch on/off logging function
FTSWHLOG	View information on log entries

openFT records the results of all file transfer requests, irrespective of whether the initiative is in the local or the remote system (outbound and inbound requests, respectively). The information on each successfully completed or aborted request is recorded in an FT logging record. The file consisting of these logging records thus represents a complete, uninterrupted documentary record of FT operation over a prolonged period of time.

FT logging can be activated or deactivated at any time with FTMODOPT.

openFT writes the logging records into the log file <inst>.SYSLOG under the OPENFT QUALIFIER of the openFT instance. The net size of he SYSLOG file depends on the number of logging records it contains. As one of your duties as FT administrator, you should regularly create backups of logging-records (the frequency depends on the volume) as a printout or tape file, for example and then delete the logging records with the FTDELLOG command. In this way you have a complete, uninterrupted log at your disposal for documentation purposes, while at the same time no storage capacity is wasted. Bear in mind the assigned file size does not change, but the space formerly occupied by the records you delete is released within the file.

The information content of the FT logging records includes:

- date and time of request processing,
- an acknowledgment indicating correct completion of a request, or the reason for request rejection or abort,
- the direction of file transfer,
- the name of the partner system involved in file transfer.
- TSN and user ID of the request initiator for requests submitted in the local system; only *REMOTE is entered for remote request initiators,
- the user ID under which the request was handled or should have been handled,
- the name of the file.
- if an abort occurs, additional information on the cause.

The FT administrator can use the FTSHWLOG command to output all FT logging records of his/her system.Two formats are available for the output: a format that is suitable for listings, and a format that is optimized for further processing. The FT administrator can also choose between a short form and a long form, i.e., a brief overview or a long detailed output.

If the FTAC functionality is being used, the logging records relevant for FTAC are saved in the same file. A detailed description of the command FTSHWLOG can be found on page 311ff; the output is presented starting on page 320.

3.6.2 The openFT job log

Beside the log file the openFT job log also contains information which may be useful for the FT administrator. Some messages are output **only** to the openFT job log; often, however, the chronological order of the messages contained in the job log is useful in the diagnosis of errors during FT operation. The information contained in the openFT job log is described in the appendix on page 414.

3.6.3 Console messages for automatic monitoring

Messages are usually issued as responses to administration commands. There are, however, also some messages which are not (or not exclusively) issued by administration commands. These messages are described in the User Guide “Messages” (on-line version only). When errors occur on accessing the request queue or the partner list, openFT generates normal system error messages.

To support automatic monitoring, some events which are not direct responses to user input are reported by openFT via a console message. Depending on which events are involved, further actions can then be initiated by NetView, for example.

The console messages for automatic monitoring occupy the message code range from FTR0300 to FTR0399. They can be activated and deactivated with FTMODOPT CONSOLE-TRAPS=*ON/*OFF. openFT outputs these messages asynchronously. This means that the output is also dependent on the settings for asychronous messages in the PARM library (see “Structure of the PARM member” on page 52ff).

Messages for monitoring partner systems

FTR0301 OPENFT: Partner '(&00)' entered state NOCON

FTR0302 OPENFT: Partner '(&00)' entered state ACTIVE

FTR0303 OPENFT: Partner '(&00)' entered state LUNK

FTR0304 OPENFT: Partner '(&00)' entered state RUNK

FTR0305 OPENFT: Partner '(&00)' entered state INACT

FTR0306 OPENFT: Partner '(&00)' entered state AINACT

FTR0307 OPENFT: Partner '(&00)' may be unreachable

FTR0308 OPENFT: Partner '(&00)' does not allow more inbound requests

FTR0309 OPENFT: Partner '(&00)' added

FTR0310 OPENFT: Partner '(&00)' removed

FTR0311 OPENFT: Partner '(&00)') entered state LAUTH

FTR0312 OPENFT: Partner '(&00)' entered state RAUTH

FTR0313 OPENFT: Partner '(&00)' entered state DIERR

FTR0314 OPENFT: Partner '(&00)' entered state NOKEY

FTR0315 OPENFT: Partner '(&00)' entered state IDREJ

Messages for monitoring openFT

FTR0320 OPENFT: abnormal termination initiated

FTR0360 OPENFT: openFT control process started

FTR0361 OPENFT: openFT control process terminated

Messages for monitoring the request queue

FTR0330 OPENFT: Request queue 85 percent full

FTR0331 OPENFT: At least 20 percent of request queue unoccupied

Messages for monitoring requests

FTR0340 OPENFT: Transfer '(&00)' successfully completed

FTR0341 OPENFT: Transfer '(&00)' terminated with error

3.6.4 Monitoring with openFT

openFT provides the option of monitoring and recording a range of characteristic data for openFT operation. The data falls into three categories:

- Throughput, e.g. total network throughput caused by openFT
- Duration, e.g. processing time for asynchronous jobs
- State, e.g. number of requests currently queued

You must be an FT administrator in order to activate, deactivate or configure monitoring.

As soon as monitoring is activated, any user can call up the data and output it based on certain criteria.

3.6.4.1 Configuring monitoring

You configure monitoring using the FTMODOPT command and the MONITORING= operand (see page 254). The following options are available:

- Activating and deactivating monitoring
- Selective monitoring based on the partner type
- Selective monitoring based on the request type

Once you have chosen your settings, they are retained until you change them explicitly. This means that they are also not changed if you reboot the computer.

You can check the current settings with FTSHWOPT. The MONITOR row indicates whether monitoring is activated and shows any criteria used for selection.

3.6.4.2 Showing monitoring data

If monitoring is activated. the monitoring data can be called up on the local system or from a remote system.

Outputting monitoring data on the local system

Use the command FTSHWMON to show monitoring data locally (see page 327).

FTSHWMON outputs the monitoring data in the form of tables that you can further process as required either programmatically or using an editor.

When you call FTSHWMON, you can select specific monitoring data for output, whether or not output is formatted and the time interval at which output is performed. You can also specify the output medium. You can find details on the values output on page 330.

Showing monitoring data on remote Unix or Windows systems

The monitoring data can also be shown in the openFT Monitor on a remote Unix or Windows system. To do this, you set up a special admission profile in your z/OS system that is specified when the openFT monitor is called and causes only the monitoring values to be read and transferred. The admission profile uses the keyword *FTMONITOR as a preprocessing command and is set up as follows:

```
/FTCREPRF NAME=MONITOR,TRANSFER-ADMISSION=ONLYFTMONITOR -
       ,FILE-NAME=*EXPANSION('|*FTMONITOR ') -
       ,FT-FUNCTION=(*TRANSFER-FILE,*FILE-PROCESSING)
```

ONLYFTMONITOR is the (freely selectable) FTAC transfer admission that must be specified when the openFT Monitor is called. Alternatively, this transfer admission can also be specified in an ft or ncopy command used to transfer monitoring data in a Unix or Windows system.

You will find details in the openFT manuals "openFT V11.0 for Unix Systems - Installation and Administration" and "openFT V11.0 for Windows Systems - Installation and Administration".

3.7 Administrating and controlling FTAC functions

In openFT for z/OS- as in the other openFT products - the entire openFT operation, i.e. execution of all file transfer and file management functions for every user ID, can be monitored and controlled with FTAC.

The FTAC commands can be used by all FT users. FTAC administrators have extended rights not available to normal users when executing these commands.

That means:

– FT users can modify their own admission sets - within the limits set by the FTAC administrator. Also, FT administrators can create and process admission profiles for their own user identification.

– For external user IDs, the admission sets and admission profiles must be administered by the FTAC administrator.

The FTAC administrators of an openFT instance are defined by means of an entry in the FTACADM member of the FT parameter library PARM.The FTAC file SYSFSA is automatically created in order to store FTAC administration data, such as admission sets, admission profiles, etc. (see section “Internal openFT data sets” on page 424).

3.7.1 Creating a default admission set

The FTAC administrator must first determine an average protection level for the user IDs in his system and use this information to modify the default admission set, whose values after the installation of openFT-AC are all 0. In the default admission set, the settings are made for the "average" FT user in the system. This provides adequate protection for most users. These specifications are valid for all user IDs which do not have their own admission set. Furthermore, in each admission set, the entry *STD can be used in different places to refer to the default admission set. This has the advantage of automatically incorporating any modification of the default admission set into these admission sets.

The FTAC administrator can set individual values for user IDs whose protection requirements deviate from the average.

3.7.2 Administrating admission sets

For the administration of admission sets, openFT-AC offers the FTAC administrator the following commands:

FTMODADS	Modify admission sets
FTSHWADS	Show admission sets

Please remember: a maximum security level is specified in the admission set for each of the six basic functions. The user ID with this admission set can use this basic function with all partner systems who have this security level or lower.

The FTAC administrator can view admission sets with the command FTSHWADS (see page 304). He can modify the admission sets with the command FTMODADS (see page 248).This command is used to modify the default admission set as well as to customize the settings for individual user IDs. The specifications of the FTAC administrator are the MAX-ADM-LEVELS in the admission set for the corresponding user ID. The user can increase the degree of protection within these levels (MAX-USER-LEVELS).

With an openFT request (outbound and inbound), the admission is compared with the FTAC security level of the partner concerned (see also page 111).

3.7.3 Administrating admission profiles

For the administration of admission profiles, openFT-AC offers the FTAC administrator the following commands:

FTCREPRF	Create admission profile
FTDELPRF	Delete admission profile
FTMODPRF	Modify admission profile
FTSHWPRF	Show admission profile

The FTAC administrator has the option of modifying foreign admission profiles:

- The administrator can create admission profiles for foreign users with the FTCREPRF command (see page 212). However, certain restrictions apply (see page 128).
- He can view them with the command FTSHWPRF (see page 346). The transfer admission of an admission profile is not output. This means that the FTAC administrator does not have access rights to the files of foreign user IDs.
- He can delete them with the command FTDELPRF (see page 238). This is the most radical of all options which should only be used in extreme cases and with good reason and upon consultation with the owner of the profile.

- He can privilege them with the command FTMODPRF (see page 273), or conversely revoke privileges.
- He can also modify them with FTMODPRF. Access to the admission profile will then be blocked if the FTAC neither possesses the SU privilege nor specifies the complete USER-ADMISSION.

Creating admission profiles for foreign user IDs

When the FTAC administrator wants to create an admission profile for a foreign user by means of the FTCREPRF command (see page 212), he can proceed in the following two ways:

- If the FTAC administrator possesses the SU privilege (see page 62), then he may set up admission profiles for other user IDs without restriction even if he does not know the current user password. The FTAC administrator may specify a TRANSFER-ADMISSION in these profiles. This can be used in FT requests immediately after being set up. Please note that FTAC administrators who possess the "SU privilege" can gain access to the files belonging to any and all user IDs by setting up the corresponding admission profiles and may therefore be able to by-pass protection mechanisms!
- Provided the FTAC administrator (without the SU privilege) knows all the data required for the USER-ADMISSION (i.e. user ID, account number and password) and specifies them when creating the admission profile, it is also possible to specify a TRANSFER-ADMISSION, with which a valid admission profile is created, i.e. the profile can immediately be used in file transfer and file management jobs.

  The password is stored as a part of this type of admission profile, so if a user changes his password, the admission profile also has to be changed.

  Example

  The FTAC administrator creates a valid admission profile for *USER1*. To do so, the administrator needs to enter the user's account number (*123456*) and password (*PASSWD1*).

  ```
  FTCREPRF NAME=HISPROF1, TRANS-ADM=READYFORUSE, -
  USER-ADM=(USER1,123456,PASSWD1)
  ```

- The FTAC administrator can also create an admission profile for a foreign user that does not contain the user's password. (When an FT job refers to this type of profile, FTAC enters the z/OS password currently valid for the user ID. That way the admission profile will not have to be changed should the z/OS password ever be modified.)

  In this case, the FTAC administrator (without the SU privilege) cannot specify TRANSFER-ADMISSION when creating the admission profile. That would create a locked admission profile, i.e. the profile can only be used in file transfer and file management jobs after the user has specified a TRANSFER-ADMISSION using the FTMODPRF command (see page 273) and after completed the USER-ADMISSION data.

Example

The FTAC administrator creates an admission profile for *USER1*. For the USER-ADMISSION, he specifies only the user ID, not the account number and the password. In that case the administrator may not specify a TRANSFER-ADMISSION.

```
FTCREPRF NAME=HISPROF2, TRANS-ADM=*NOT-SPECIFIED, -
USER-ADM=(USER1,*NOT-SPECIFIED,*NOT-SPECIFIED)
```

The FTAC administrator views the admission profile using the FTSHWPRF command (see page 346). The short output shows that the profile is locked (indicated by the "!" in front of the profile name):

```
FTSHWPRF NAME=HISPROF2, SEL=(OWNER=*ALL)
  OWNER       NAME
  USER1      !HISPROF2
```

The long output shows that no valid TRANSFER-ADMISSION was specified in the profile:

```
FTSHWPRF NAME=HISPROF2, SEL=(OWNER=*ALL), INF=*ALL
 HISPROF2
  TRANS-ADM   = (NOT-SPECIFIED)
  USER-ADM    = (USER1,NOT-SPECIFIED,NOT-SPECIFIED)
  PROC-ADM    = SAME
  FT-FUNCTION = (TRANSFER-FILE, MODIFY-FILE-ATTRIBUTES,
                 READ-FILE-DIRECTORY)
```

The user now assigns a TRANSFER-ADMISSION and supplements the USER-ADMISSION data:

```
FTMODPRF NAME=HISPROF2, TRANS-ADM=NOWREADYFORUSE, -
USER-ADM=(USER1,123456,PASSWD1)
```

Now the admission profile can be used in file transfer and file management jobs as well.

The user views the admission profile with the FTSHWPRF command (see page 346).

The short output shows that the profile is no longer locked:

```
FTSHWPRF NAME=HISPROF2
  OWNER       NAME
  USER1       HISPROF2
```

The long output shows that the user's account number has been included in the admission profile along with the identifier YES for the USER-ADMISSION password:

```
FTSHWPRF NAME=HISPROF2, INF=*ALL
 HISPROF2
  USER-ADM    = (USER1,123456,YES)
  PROC-ADM    = SAME
  FT-FUNCTION = (TRANSFER-FILE, MODIFY-FILE-ATTRIBUTES,
                 READ-FILE-DIRECTORY)
```

Privileging admission profiles

In exceptional cases, the FT user can use a privileged admission profile to disregard the specifications of own admission profile. The user ID protection is maintained in this case, by the fact that only very restricted access is permitted into the admission profile. Exceptional cases where this is allowed include:

- if a particular file needs to be transferred,
- if follow-up processing is not permitted or severely restricted,
- if a partner system with a higher security level is permitted to carry out file transfers with the user ID, but others with lower security levels are not.

The procedure to follow when privileging an admission profile is simple:

1. The user creates an admission profile for the planned task with the command FTCREPRF.
2. The FTAC administrator views the admission profile with the command FTSHWPRF to determine if the profile presents a threat to data security.

Example

```
FTSHWPRF NAME=PROFPROD,
         SELECT-PARAMETER=(OWNER-IDENTIFICATION=DONALD), -
         INFORMATION=*ALL
```

Short form:

```
FTSHWPRF PROFPROD,SEL=(,DONALD),INF=*ALL
```

The output has the following form:

```
PROFPROD
 IGN-MAX-LEV = (IBR)
 FILE-NAME   = PROFIT
 USER-ADM    = (DONALD,M4711DON,OWN)
 PROC-ADM    = SAME
 SUCC-PROC   = NONE
 FAIL-PROC   = NONE
```

The first line of the output shows the name of the admission profile, the second line the values which Donald has set in the command FTCREPRF (see page 212) or which are determined by the default values, if Donald doesn't set them himself.

3. If the profile will not endanger security, the FTAC administrator privileges it with the help of the command FTMODPRF.

   Example

   ```
   FTMODPRF NAME=PROFPROD,
            SELECT-PARAMETER=(OWNER-IDENTIFICATION=DONALD), -
            PRIVILEGED=*YES
   ```

In a privileged admission profile, only the transfer admission and the parameter PRIVILEGED may be modified by the user. This prevents the misuse of any profiles, once privileged.

3.7.4 Transfer FTAC environment - the environment functions

The following commands are available for the environment functions:

FTEXPENV	export FTAC environment to export file
FTIMPENV	import FTAC environment from export file
FTSHWENV	show FTAC environment from export file

The FTAC administrator can have admission profiles and sets written (i.e. “exported”) to a file and thus back up all admission profiles and sets that exist on the computer.
In addition, this function is useful when a user migrates from one computer to another. In this case, the FTAC administrator first backs up the existing FTAC environment to a file and then re-installs this on another computer. The FTAC user can then continue to work in the same FTAC environment as before, i.e. with the same admission profiles and the same admission set. Depending on the rights of the FTAC administrator who is performing the import and the security settings in the "import system", it may be necessary to set up privileges explicitly on the new computer and release the transfer admissions explicitly.

The FTAC administrator can also selectively back up (FTEXPENV, page 241) admission sets and profiles by using corresponding parameter specifications and then restore them when needed (FTIMPENV, page 245). This can be done with:

– admission profiles and admission sets of one or more users (up to 100)

– all admission profiles and admission sets on a given computer

– only admission sets, no admission profiles

– only admission profiles, no admission sets

The contents of a backup file can be viewed with the command FTSHWENV (see page 308).

Example

Donald Duck needs to work on a new computer under the same user ID DONALD. Donald would like to keep the same admission set and admission profiles as before. To do this, the FTAC administrator Dagobert backs up the admission set and the admission profiles for the user ID DONALD in the file DONALD.FTAC.BKUP.

```
FTEXPENV TO-FILE=DONALD.FTAC.BKUP,USER-IDENTIFICATION=DONALD
```

Being a conscientious FTAC administrator, Dagobert Duck checks if the desired backup is in the file DONALD.FTAC.BKUP.

```
FTSHWENV FROM-FILE=DONALD.FTAC.BKUP
```

He receives the following output:

```
              MAX. USER LEVELS               MAX. ADM LEVELS         ATTR
 USER-ID  OBS  OBR  IBS  IBR  IBP  IBF   OBS  OBR  IBS  IBR  IBP  IBF
 DONALD     1    1    0    1    0    0     1    1    0    0    0    0
 OWNER     NAME
 DONALD   *PROFPROD
```

Now, Dagobert transfers the file DONALD.FTAC.BKUP to the user ID of the FTAC administrator on the new computer.
There, Sylvester the Cat, the FTAC administrator for the new computer, transfers the admission set and the admission profiles of the user ID DONALD from the file DONALD.FTAC.BKUP.

Sylvester is also a conscientious administrator. He checks if Donald´s admission sets and profiles are a threat to the security of his system (he doesn’t trust Dagobert in the slightest):

```
FTSHWENV FROM-FILE=DONALD.FTAC.BKUP
```

and he receives the same output as above.

Then Sylvester imports Donald´s admissions from the file DONALD.FTAC.BKUP onto his system:

```
FTIMPENV FROM-FILE=DONALD.FTAC.BKUP
```

Sylvester the Cat must then privilege Donald’s profile

```
FTMODPRF PROFPROD,,(,DONALD),PRIV=*Y
```

Finally, Donald must release the imported profiles before he can work with them.

```
FTMODPRF NAME=*ALL,TRANSFER-ADMISSION=*OLD(VALID=*YES)
```

3.7.5 The FTAC logging function

openFT-AC checks the access rights of every FT request which the protected system is involved in and logs the results. This information is stored in the so-called FTAC logging records.
The following information can be called up by the FTAC administrator:

- logging date
- type of logging record (FT or FTAC logging record)
- logging number of the FT request
- time of access check
- code for the function of the FT request
- reason for a possible job rejection by FTAC in the form of a return code (you can view the meaning of the return code with the FTHELP command)
- transfer direction of the FT request
- name of the partner system with which the FT request was/is to be carried out
- LOGON authorization (USER-IDENTIFICATION) of the initiator of requests which were made in the local system (or *REMOTE for remote request initiators)
- name and privileging identifier of any admission profiles used
- the local file name

FTAC only checks the admission for a request on the basis of the admission sets and admission profiles. openFT logs whether or not it can actually execute the request in the FT or ADM log records. FT log records are not written for all transfer requests. If, for instance, FTAC rejects a transfer request as a result of a negative access check, only an FTAC log record exists, and no FT log record. For further details, see section “FTSHWLOG Display logging records” on page 311.

It is not possible to completely deactivate output of FTAC log records (not even with the FTMODOPT command, that can deactivate FT log records).AES (Advanced Encryption Standard) is used as the encryption method.

The FT command FTSHWLOG (see page 311) can be used by the FTAC administrator to find out about all access checks which have been carried out by openFT-AC to date. This facilitates processes such as system inspections.

Codes for the function of the FT request

The entries in front of the brackets indicate the log representations of the individual FT functions. The FT requests themselves can consist of groups of FT functions. However, only one will appear in the logging record. These groups are listed in the brackets.

␣	TRANSFER-FILE	(WRITE-FILE + ... or READ-FILE + ...)
A	READ-FILE-ATTRIBUTES	(READ-FILE-ATTRIBUTES + ...)
D	DELETE-FILE	(DELETE-FILE + ...)
C	CREATE-FILE	(CREATE-FILE + ...)
M	MODIFY-FILE-ATTRIBUTES	(MODIFY-FILE-ATTRIBUTES + ...)
R	READ-DIR	(READ-DIR + ...)
CD	CREATE-DIR	
MD	MODIFY-DIR	
DD	DELETE-DIR	
L	FTP-LOGIN [1]	

[1] Generated on failed access attempts via openFT-FTP

To make the output of the command FTSHWLOG provide more of an overview, you can specify values or value ranges for various output parameters when calling up the command. This permits you to be selective in the output of logging records

Deleting logging records

FT and FTAC administrators are the only users in the system who can view and delete all FTAC log records. The FT command used for this is FTDELLOG (see page 234). The FTAC user cannot use this function.

FTAC logging records can only be deleted from the oldest date up to a specified date. This ensures that there will be no gaps in the log file up to the most current record.

In theory, openFT-AC can write any number of logging records ("until the disk is full"). From time to time, the FTAC administrator should make a backup of existing logging records (either print out a hard copy or make a copy on tape) and then delete these logging records from the log file. This ensures that the logging records will provide a continuous record over an extended period of time, as well as prevent the log file from getting too large.

3.8 Using openFT in a SYSPLEX cluster

In openFT you can run multiple openFT instances on one computer simultaneously. Because of these instances, should a computer fail, you are in a position, for example, to carry over the functionality of the openFT to another computer of a SYSPLEX configuration, which is already running openFT.

After installing openFT, the **default instance** exists on each computer. When instances are displayed (FTSHWINS), the default instance is always displayed first.

Up to 16 additional instances can be created by administration. Each of these instances, including the standard instance, consists of the following components:

- The request and request file SYSRQF, the partner list SYSPTF, the logging file SYSLOG, trace files, options file SYSOPF, and the FTAC profile file SYSFSA.
- Each instance requires its own network address; this always remains the same, independent of the real host. Therefore, the name of the host on which an instance is running is specified in the FJGEN command. This host name must always be accessible under the same network address.

The openFT installation files are only available once per computer and are shared by all the instances. The same version, however, must be installed on all the computers in the cluster (openFT version, proofing version, PTFs, etc.).

openFT commands that are called during a preprocessing, postprocessing or follow-up processing session, run under the same instance as the request that initiated the processing.

3.8.1 Setting up openFT instances

Instances are created by means of the FJGEN command (see page 48). They are identified and administered via the instance name that you specify in INSTANCE NAME during FJGEN processing. For the sake of clarity, the instance name should be a name part of all the openFT files and libraries that belong to the corresponding instance (e.g. FTAC files etc.).

WARNING!
The instance name should not be confused with the so-called instance identifier that is defined using the IDENTIFICATION parameter in the FTMODOPT command. As of openFT V8.1, the instance identifier is used by partner systems in order to authenticate your openFT instance. Similarly, you need these partner systems' instance identifiers in order to authenticate them in the local system.

If you are only working with one instance then you should use the standard instance STD. This name is also proposed as the default in FJGEN.

Instance-specific CONN file

There is a so-called CONN file associated with each instance. It contains information required for internal communication between the command client from the library <openft qualifier>.OPENFT.NCLOAD and openFT from the library <openft qualifier>.OPENFT.LOAD and for encrypting this communication.

If you want to work with a specific instance then before you call any openFT functions, the instance-specific CONN file must be allocated by:

```
<openft qualifier>.<inst>.CONN
```

This is possible, for example, using the following call:

```
ALLOC DSNAME('<openft qualifier>.<inst>.CONN') DDNAME(OPENFT) SHR REUSE
```

Where <openft qualifier> and <inst> correspond to the OPENFT QUALIFIER and INSTANCE NAME specifications in the FJGEN command.

It is **urgently recommended** that you allocate the CONN file before calling the openFT command. This also applies if only the default instance exists!

Instance-specific NCLOAD

To allow openFT commands to be called under TSO or from a CLIST, the NCLOAD <openft qualifier>.OPENFT.NCLOAD must be entered in the search path/sequence for TSO commands. This can be done using the following command, for instance:

```
TSOLIB ACT DATASET(<openft qualifier>.OPENFT.NCLOAD)
```

Instance-specific CLIST

To administer openFT, it is also necessary to concatenate the instance-specific CLIST <openft qualifier>.<inst>.CLIST (either in the current TSO session or by incorporating it in the LOGON procedure, see page 40). This also applies to the standard instance.

If multiple openFT instances are to run in parallel on a computer under the same user ID, then different job names must be set in the FJBATCH members of the instance-specific CLISTS (for example, USERAX instead of USERAF). These are the batch jobs that load the appropriate openFT instances.

Exchange settings between instances

It is a simple matter to exchange partner entries between the instances using the LAYOUT=*ZOS-PROC parameter in the FTSHWPTN command (see the example for the FTSHWPTN command on page 352). FTAC components can be taken over using the commands FTEXPENV and FTIMPENV.

Show information about instances

You can use the FJGENPAR command to view the installation parameters of the current instance during operation (and modify them, if required, by means of a new FJGEN run). FTSHWINS allows you to obtain information on the known openFT instances running on a computer, provided that openFT has been started as a subsystem.

3.8.2 Importing an instance to another computer

The following steps are required to change over an openFT instance to another computer:

- Stop the instance on the original computer (FTSTOP).
- Unload the instance on the original computer (FTTERM). This unlocks all of the files required by openFT (request file, transfer files, etc.).
- Import the variable files, the network address and all of the files required by the requests to the destination computer. This can contain, among other things, the switching over of one or several pubsets).
- Load the instance on the destination computer (FJINIT).
- Start the instance on the destination computer (FJSTART).

After importing an instance to another computer, openFT finishes the (under some circumstances restartable) requests, whose admissions were already checked before importing. The new environment must have the same prerequisites as the old computer (the same IDs with the same file access admissions).

All file systems that are accessed by requests must be available. All requests whose file systems are not accessible during restart attempts are aborted.

On the new computer, the network view must be the same as that on the old computer. This means that the same host names for partner computers must be available and they must refer to the same partner computer. The network address of the host on which the instance is running, must be seen from the outside the same as from the address of the host, on which the instance was previously running.

The standard instance STD cannot be switched.

3.9 Diagnostics

By way of support for error diagnosis, you can use FTMODOPT TRACE=*ON/*OFF to activate and deactivate the FT trace monitor. This function can be switched on and off irrespective of whether the FT system is active or inactive.

3.9.1 Controlling the trace function

The FT administrator uses the following commands to control the trace function:

FTMODOPT	Modify operating parameters
FTMODPTN	Modify partner characteristics
FTSHWOPT	Information about operating parameters
FTSHWPTN	Information about partner systems

You can set the scope of openFT traces globally to apply to multiple levels or individually for specific partners. You may also suppress the majority of trace entries for selected partners. If you do this, you can then only view those entries which were created before openFT identified the partner system.

The following table shows four typical applications for trace operation.

FTMODOPT	**FTMODPTN**	**Task**	**Effect**
TRACE=*ON	TRACE= *BY-FT-OPTIONS	General tracing of FT operations.	FT operation is fully traced.
TRACE=(SWITCH=ON, OPTIONS= NO-BULK-DATA)	TRACE= *BY-FT-OPTIONS	Connect tracing for all openFT partners.	Mass data transfers are not recorded. Recommended for long-lived traces.
TRACE=(SWITCH=ON ,PART-SELECTION= *FTP)	TRACE= *BY-FT-OPTIONS	Tracing of a a certain type of partner over an extended period. (here, ftp partners)	All events relating to a selected partner type are logged. Despite the extended period, the trace volume does not become excessive.
TRACE=(SWITCH=ON ,REQ-SELECTION= *REM)	TRACE= *BY-FT-OPTIONS	Tracing of a specific type of request (here, requests submitted by a remote system)	All events relating to certain request types are logged. Despite the extended period, the trace volume does not become excessive.

The following table indicates the interrelations between the most important FTMODOPT and FTMODPTN trace settings.

FTMODOPT	FTMODPTN	Effect
TRACE=*OFF	equals	*OFF
TRACE=*ON	TRACE=*BY-FT-OPTIONS	*ON
	TRACE=*UNCHANGED	Setting retained
	TRACE=*ON	*ON
	TRACE=*OFF	*OFF
TRACE=(SWITCH=ON, PARTNER-SELECTION= partner type)	TRACE=*BY-FT-OPTIONS	*ON if suitable partner type *OFF if unsuitable partner type
	TRACE=*UNCHANGED	Setting retained
	TRACE=*ON	*ON
	TRACE=*OFF	*OFF
TRACE=(SWITCH=ON, REQUEST-SELECTION= request type)	TRACE=*BY-FT-OPTIONS	*ON if suitable request type *OFF if unsuitable request type
	TRACE=*UNCHANGED	Setting retained
	TRACE=*ON	as *BY-FT-OPTIONS
	TRACE=*OFF	*OFF

You will find details on the trace files and the way they are formatted with FTTRACE in the section “Format of the trace files” on page 419 and in the section “FTTRACE command” on page 420.

3.9.2 Diagnostic records

If, despite due care and attention, an error occurs that neither the FT administrator nor the z/OS system administrator can rectify, contact your Service Center. To facilitate troubleshooting, please submit the following:

- detailed description of the error situation and statement indicating whether the error is reproducible
- trace files
- if applicable the result list of the request that triggered the error
- job list of the openFT job (also from partner system is possible)
- general information as for z/OS system error
- version of the FT partner and details of the transport system (e.g. DCAM, CCP / CMX, VTAM, etc.)
- openFT dump files with the name
  '<openft qualifier>.<inst>.SYSFDF.Dyymmdd.Thhmmss'
  or the SYSUDUMP assigned in the FJBATCH- job
- If necessary, create an HPNS trace if problems arise with respect to TCP/IP. To do this, you must create a member DIAGPAR in the instance-specific PARM library and restart the batch job. See DIAGPAR member in the FT parameter library.

DIAGPAR member in the FT parameter library

To allow additional diagnosis with an HPNS trace, the member DIAGPAR must be supplied with the following values in the instance-specific FT parameter library PARM:

HPNSTRACE=17825791

DIAGSTAMPS=12

If the openFT (batch job) is restarted with these entries, additional diagnosis records on data communication are returned at the socket interface or written to the file <inst>.SYSUDUMP in the event of a dump written to the file <inst>.SYSUDUMP.

3.10 Backing up the configuration data

You should back up the configuration data of your openFT instance at regular intervals. This ensures that you will be able to restore openFT operation with as little delay as possible using the original runtime environment after a computer has failed or been replaced, for instance.

You should always store the operating parameter settings, the partner list and, where applicable, the FTAC environment in backup files. To do this, you can proceed as follows (the filenames are only examples and the backup files must not already exist):

- Backing up the operating parameter settings:

```
FREE DDNAME(SYSPRINT)

ALLOC DSNAME(OPTZOS.CLIST) DDNAME(SYSPRINT) NEW KEEP DSORG(PS) RECFM(F,B)
LRECL(80)

FTSHWOPT OUT=*STDOUT(*ZOS-PROC)

FREE DDNAME(SYSPRINT)
```

- Backing up partner list entries:

```
FREE DDNAME(SYSPRINT)

ALLOC DSNAME(PARTZOS.CLIST) DDNAME(SYSPRINT) NEW KEEP DSORG(PS) RECFM(F,B)
LRECL(80)

FTSHWPTN OUTPUT=*STDOUT(*ZOS-PROC)

FREE DDNAME(SYSPRINT)
```

- Backing up the FTAC environment:

```
FTEXPENV FTAC.SAVE
```

4 Menu interface for the FT administrator

This chapter describes the easy-to-use menu interface via which you can perform your FT and FTAC administrator tasks. The previous distinction between one menu system for administrators and another for users has been discarded and both now see the same entry menu (Primary Option Menu). Naturally, FT and FTAC administrators have more rights.

If you are administering openFT under TSO (as opposed to controlling openFT via an operator console or via NetView, described on page 385 or page 387 respectively), you can use special FT administrator commands. These commands are described in the chapter “Command interface for the FT administrator” on page 165.

In MVS systems in which the product ISPF is installed, however, you have the option of using the menu interface (ISPF) described below, with the usual choice of menus and data entry panels, immediate warnings in the event of errors, help functions etc. You can use this menu interface to

- load and start the openFT load module,
- modify the request file and the partner list,
- activate the openFT instance,
- control the use of resources,
- request information about the openFT instance,
- monitor the openFT instance,
- request information on FT requests,
- cancel/abort FT requests,
- deactivate and reactivate file transfer requests submitted in the local system to individual remote FT systems,
- deactivate the openFT instance,
- terminate the openFT load module.
- administer the local keys of an openFT instance
- administer FTAC admission sets and FTAC admission profiles if required.

The menu interface is therefore a more user-friendly means of executing functions that can also be executed via the command interface.

Exceptions

You can execute the following functions only via the command interface:

- For the FJGEN command for setting the form installation parameters (see page 191) the menu interface does not provide a function which corresponds to this command; you must enter the command directly in order to perform the corresponding installation step (see section “Setting openFT installation parameters with FJGEN” on page 48). The parameters you set in this installation step, however, can be displayed at any time via a menu.

In contrast, the menu interface provides you with a direct means of editing the members PARM, TSOJOB, JCLJOB, PRTJOB, TSOVVJOB, TSONVJOB, TSOVFJOB, SUCCMSG, FAILMSG and TNSTCPIP of the FT parameter library (see page 49). The FT parameter library itself, however, must already exist and its name must have been specified when setting the installation parameters with FJGEN. (You cannot process the member FNAMECTB in the FT parameter library via the menu interface.)

4.1 Software requirements

In order to use the menu interface for the FT administrator, you must first ensure that the IBM program product "Interactive System Productivity Facility" (ISPF) is installed on your system.

The installation of the libraries with the openFT panel definitions, CLISTs and messages is described in the section “Making the commands and the ISPF panels available” on page 40.

4.2 Setting an openFT instance

On a z/OS system 16 so-called openFT instances may be present in parallel on a z/OS system. In themselves, these instances represent complete openFT systems each with their own request file and own partner list, their own addresses and, in some cases, their own FTAC settings.

In addition to the specific files of the openFT instance that you want to use, (CLIST and possibly CONN file), the corresponding libraries must also be concatenated to permit use of the menu interface:

– the CLIST OPENFT.PANEL.CLIST
– the panel library OPENFT.PANELS
– the message library OPENFT.PANEL.MSG

In a running TSO session, you can perform these allocations manually. This means, for example, that you can also change the openFT instance within the TSO session. You should store the necessary commands in a CLIST and execute these in TSO mode. You can also record the name of this CLIST in a LOGON procedure so that the commands are always executed when you log on.

Example of this type of CLIST

```
allocate file(sysproc) dataset('isp.sispclib' -
'OPENFTQU.STD.CLIST' -
'USERA.OPENFT.PANEL.CLIST' -
'nix1.ispf.isrclib') reuse shr
allocate file(ispplib) dataset('isp.sisppenu' -
’USERA.OPENFT.PANELS' 'nix1.ispf.isrplib') reuse shr
allocate file(ispmlib) dataset('isp.sispmenu' -
'USERA.OPENFT.PANEL.MSG' 'nix1.ispf.isrmlib') reuse shr
ALLOC DSNAME('OPENFTQU.STD.CONN') DDNAME(OPENFT) SHR REUSE
```

4.3 Representation and utilization

You call the initial panel of the menu interface (Primary Option Menu, i.e. FTMAIN panel for openFT without FTAC or FTACM panel for openFT with FTAC) under TSO as follows:

```
EXECUTE '<openft qualifier>.OPENFT.PANEL.CLIST(FJMENU)'
```

Further information is given in the section “Making the commands and the ISPF panels available” on page 40.

The structure of the menu system is described below. Detailed help on each individual panel can be found online by pressing the F1 key.

Terminal operation is subject to the rules that usually apply with IBM ISPF:

- The ENTER key causes terminal input to be passed on and, where appropriate, verified. The particular reaction that follows depends on the panel currently displayed.
- In many cases, a data entry panel appears in which you can or must make entries. An action is then executed which has the same effect as issuing the corresponding FT command. In the next step, the message issued by openFT in response to this action is displayed on the screen. You exit this display in the usual manner using END, RETURN or the "jump function" (see below).
- The END command causes a return to the panel preceding the current panel in the panel hierarchy. In this case, no action is usually executed. This enables you to cancel actions which you have selected by mistake.
- The RETURN command causes a return to the Primary Option Menu. In this case, too, no action is executed.
- The "jump function" of ISPF (calling a sequence of panels in one step e.g. "=p.3") is supported. In this case, the effect of the END command differs from the one described above: as usual with ISPF, "the panel preceding the current panel in the panel hierarchy" is interpreted as the panel from which the "jump" was made; so the END command causes that panel to be displayed. In this case, too, no action is executed.
- Function keys PF1 through PF12 (or through PF24) can be used as usual with ISPF.
- If a syntax error or any other type of error is detected in the input, a short message is displayed in the top right-hand corner of the screen indicating the error. At the same time, the cursor is positioned at the input field concerned. Subsequent entry of the HELP command causes a more detailed message to be displayed in the third line on the screen. Repeating the HELP command causes a help panel to be displayed.
- The HELP panels for the individual functions form a hierarchy; you can therefore use the usual commands to "browse" through these help panels (e.g. ENTER to display the next help panel, BACK to display the previous help panel, etc.).

– Data you have entered in data entry panels is generally deleted as soon as you exit the panel. Exceptions are noted as appropriate for each panel.

– The data you have entered is not deleted, however, if the same data entry panel is displayed again following execution of the function (ENTER). This is the case for a number of functions which can be effectively repeated a number of times in succession (e.g. the function ADD REMOTE SYSTEM TO NETWORK DESCRIPTION). In this case, the data you have entered is also displayed once again and you can modify it before executing the function again. This applies until you finally exit the panel using END (or RETURN or the "jump function").

– In the case of "string" type input fields, the uppercase/lowercase notation is taken over, otherwise all inputs are converted to uppercase.

– The equals sign "=" has its usual ISPF navigation function (e.g. "=x" to exit the interface). For this reason, it is not possible to pass openFT any values that start with "=" via the interface.

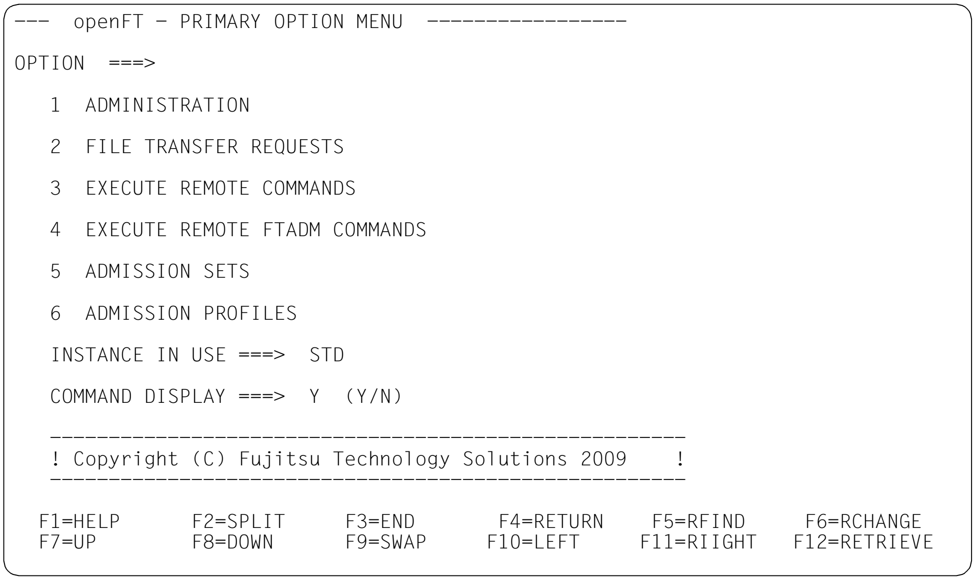
Refer to the relevant IBM manuals for further information about ISPF.

The entries you can or must make in the fields of the data entry panels correspond to the parameter values which you must specify for the corresponding FT administration command. They are described in the chapter “Command interface for the FT administrator” on page 165).

The messages issued by openFT in response to your actions are also the same as those issued at the command interface. These messages and their meanings are given in the appendix (see page 430).

openFT displays the "PRIMARY OPTION MENU" illustrated on the next page as the entry panel. Menu items 4and 5in this menu are only available if openFT-AC is installed.

PRIMARY OPTION MENU



```
---  openFT - PRIMARY OPTION MENU  -----------------

OPTION  ===>

   1  ADMINISTRATION

   2  FILE TRANSFER REQUESTS

   3  EXECUTE REMOTE COMMANDS

   4  EXECUTE REMOTE FTADM COMMANDS

   5  ADMISSION SETS

   6  ADMISSION PROFILES

   INSTANCE IN USE ===>  STD

   COMMAND DISPLAY ===>  Y  (Y/N)

   ______________________________________________________
   ! Copyright (C) Fujitsu Technology Solutions 2009    !
   ------------------------------------------------------

  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIIGHT   F12=RETRIEVE
```

This is the initial panel of the menu interface for the FT administrator if FTAC is used. It is qualified as the "Primary Option Menu", which means that it is the panel to which you return from any subsequent panel after entering the RETURN command.

You enter YES or NO in the COMMAND DISPLAY field in order to specify whether or not the FT commands which correspond to the functions you select in the subsequent menus are to be displayed on the screen, together with all the parameters which correspond to your entries in the data entry panel, if applicable.

Provided you do not change this setting, it remains valid throughout the session and is retained after the session is terminated.

The following list illustrates the hierarchy of the subsequent menus and functions that can be accessed from the Primary Option Menu. FTAC-specific items are present only if openFT-AC is installed.

1. ADMINISTRATION
    1. OPERATING PARAMETERS
        1. LOAD openFT (only as a batch job)
        2. START LOCAL FT SYSTEM
        3. STOP LOCAL FT SYSTEM
        4. TERMINATE openFT
        5. KEY MANAGEMENT
            1. CREATE KEY SET
            2. DELETE KEY SET WITH REFERENCE ...
            3. UPDATE KEY SET
        6. MODIFY FT OPTIONS
    2. REMOTE SYSTEMS

        (add, list, modify, remove FT-partners)
    3. ADDITIONAL PARTNER DEFINITIONS (EDIT TNSTCPIP)
    4. FTAC ENVIRONMENT
        1. EXPORT FTAC ENVIRONMENT
        2. IMPORT FTAC ENVIRONMENT
        3. SHOW FTAC ENVIRONMENT
    5. INSTALLATION PARAMETERS
        1. DISPLAY DIALOG ENVIRONMENT
        2. DISPLAY PRESET INSTALLATION PARAMETERS
        3. EDIT PARM (INSTALLATION PARAMETERS)
        4. PRTJOB: ...
        5. TSOJOB: ...
        6. JCLJOB: ...
        7. TSOVVJOB: ...
        8. TSONVJOB: ...

        - 9 TSOVFJOB : ...
        - S SUCCMSG: ...
        - F FAILMSG: ...
    - 6 LOGGING/DIAGNOSTIC FUNCTIONS
        - 1 SHOW LOGGING RECORDS
        - 2 DELETE LOGGING RECORDS
        - 3 SHOW DIAGNOSTIC INFORMATION
        - 4 SHOW openFT TRACE DATA
- 2 FILE TRANSFER REQUEST
    - 1 ENTER FILE TRANSFER REQUEST
    - 2 SHOW/MODIFY/CANCEL FILE TRANSFER REQUEST(S)
    - 3 SHOW LOGGING RECORDS
    - 4 SHOW ALLOWED PARTNER SYSTEMS
- 3 EXECUTE REMOTE COMMANDS
- 4 EXECUTE REMOTE FTADM COMMANDS
- 5 ADMISSION SETS
- 6 ADMISSION PROFILES

    (Create, list, modify, delete FT admission profiles)

4.4 Error messages

The messages issued by openFT in response to your actions are the same as those issued at the command interface. These messages and their meaning are given in the appendix (see page 430).

Errors you make when entering data into the panels are displayed in the usual way in ISPF (output of a short message or, if the HELP command is issued, a long message).

Short messages and long messages can also occur for other reasons, however, e.g. in the event of errors when accessing temporary files. There are the following temporary files:

<inst>.FJCMD.TMP.OUT

When some of the menu interface functions are executed, a temporary PS data set is created to buffer the command. This data set is usually deleted again after the function has been executed.

inst: Instance name of the currently set openFT instance

<inst>.FJCMD.TMP.MSG

When some of the menu interface functions are executed, a temporary PS data set is created to buffer the messages generated by openFT. The content of this data set is automatically displayed on the screen (internal call of the PDF BROWSE service). Under normal circumstances, the data set is deleted when the display is closed.

inst: Instance name of the currently set openFT instance

If a temporary file cannot be created, you receive the following messages:

Short Message: `I/O - ERROR`

Long Message: `ERROR OCCURRED ON ACCESSING TEMPORARY OUTPUT FILE.`

4.5 Calling EDIT via the menu interface

The menu interface provides you with a direct means of creating and editing (EDIT) some members of the FT parameter library (PARM, TSOJOB, JCLJOB, PRTJOB, SUCCMSG, FAILMSG, TNSTCPIP, CLASSDEF and CLASSATT). The FT parameter library itself, however, must already exist.

When creating or modifying members, please note that they may contain **no line numbering**. You must therefore set NUMBER OFF in your EDIT profile.

If you call the PDF service EDIT via the menu interface for the FT administrator, the "recovery" function is also available. This means that:

– you can use the UNDO command.

– Following a system failure during an EDIT session, the old session is first recovered the next time you call EDIT (for any member). This is indicated by the following messages:

  Short Message: `CAUTION – RECOVERY`

  Long Message: `THIS IS EDIT RECOVERY OF MEMBER...`

  You can now continue with this EDIT session. If you exit this session, the member you originally selected is edited.

  Following a system failure, you must call PDF-EDIT in the same way as before, i.e. either via the menu interface for the FT administrator or via the general ISPF/PDF interface of your system.

  Otherwise it is not possible to recover the session which was interrupted by the system failure.

4.5.1 Error messages for EDIT

If errors occur when the PDF service EDIT are called via the menu interface for the FT administrator, an "ISPF DIALOG ERROR" screen is displayed. The error messages displayed here have the following meaning:

```
DATA SET NOT CATALOGED
'.....' WAS NOT FOUND IN CATALOG.
```

Meaning
No FT parameter library exists or the file is not catalogued.

Response
Create or catalog the FT parameter library (PO or PDSE data set), observing the notes provided in the section section “Setting up the FT parameter library” on page 49).

```
AUTHORIZATION FAILED
YOU MAY NOT USE THIS PROTECTED DATA SET. OPEN 913 ABEND.
```

Meaning
You may not access the (RACF-protected) FT parameter library.

Response
Modify (or ask the administrator to modify) the access rights for the FT parameter library.

```
MEMBER IN USE
MEMBER IS BEING UPDATED BY YOU OR ANOTHER USER.
```

Meaning
The selected member of the FT parameter library is already being processed by another FT administrator.

Response
Coordinate modifications to members of the FT parameter library.

5 Central administration

Central administration in openFT covers the functions **remote administration** and **ADM traps**. openFT for z/OS supports both functions and can thus be integrated in an overall strategy.

Compared with openFT V10.0, these functions offer considerable advantages that are of particular benefit if you want to administer and monitor a large number of openFT instances. These benefits include:

- Simple configuration

  The configuration data is maintained centrally on the **remote administration server**, which means that it only exists once. The creation of roles in the form of **remote administrators** and the grouping of several instances make it possible to implement even complex configurations simply and in a clearly structured way. Subsequent changes are simple to incorporate and thus make the configuration easy to maintain.

  The remote administration server runs on either a Unix or a Windows system.

- Simplified authentication procedure

  If you wish to use authentication for reasons of security, it is only necessary to distribute a few keys:

  – For the direction to the remote administration server, the keys of computers from which remote administration is to be performed must be stored on the remote administration server.

  – For the direction from the remote administration server to the instances to be administered, it is only necessary to store the public key of the remote administration server on the openFT instances to be administered.

- High performance

  The new remote administration interface allows far longer command sequences than in openFT up to V10.0.

  It is possible to configure the remote administration server in such a way that it is available exclusively for remote administration. In this case, there is no dependency on normal FT operation and hence no mutual impact.

- Simple administration

  Remote administrators only need one (central) transfer admission. Up to openFT V10, the remote administrators had to remember the access data for each openFT instance to be administered.

- Central logging of important events

  ADM traps can be generated if certain events occur on openFT instances. These are sent to the (central) ADM trap server and stored permanently there. This allows remote administrators to evaluate important events at a later time and for specific instances.

- Compatible integration of earlier openFT versions

  Instances running versions of openFT as of V8.0 can simply be added to the configuration and administered in the same way as instances as of V11.0. All the administration functions offered by the corresponding openFT version can be used.

5.1 Remote administration

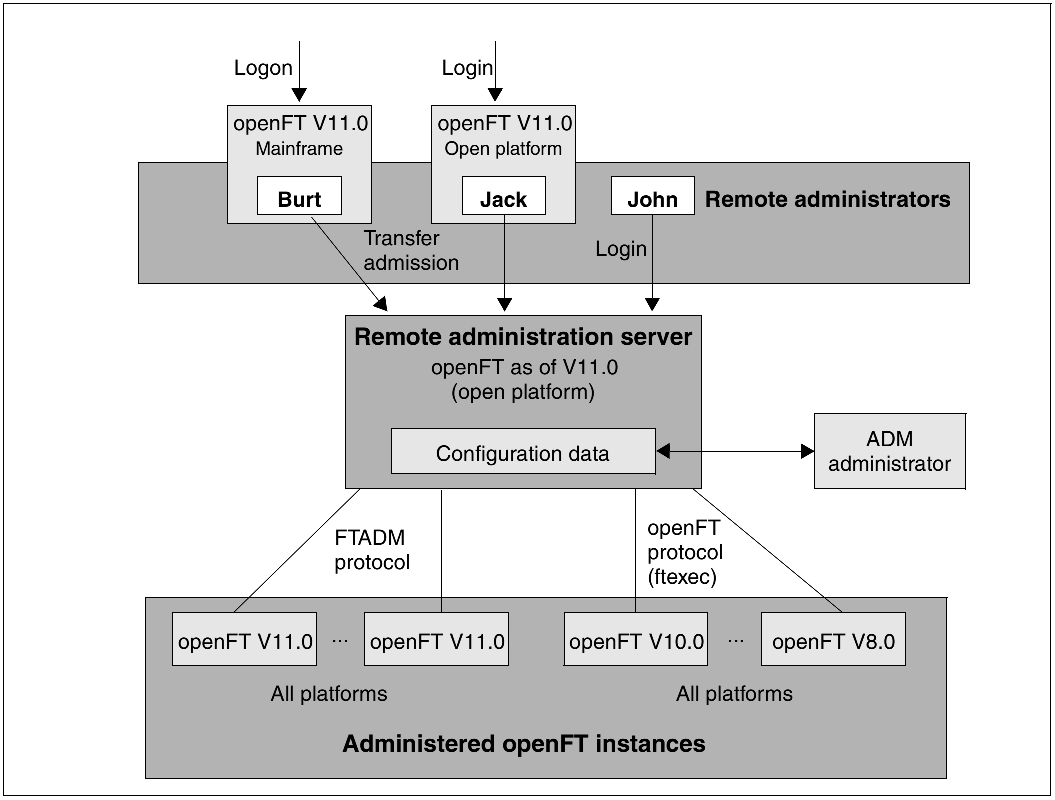
openFT allows you to set up a remote administration server via which you can administer your openFT instances on the various platforms. You can choose to use any openFT instance as an administration workstation.

This section describes:

- the remote administration concept
- how to configure an openFT instance on z/OS for remote administration
- how to enter remote administration commands on z/OS

5.1.1 The remote administration concept

The figure below shows the remote administration components and the most important configuration options on the basis of a deployment scenario.



Remote administration components

Remote administration comprises the following components:

Remote administration server

Central remote administration component. This runs on a Unix or Windows system with openFT as of V11.0 and contains all configuration data for remote administration.

Multiple remote administration servers can be defined in a complete configuration.

i You will find details on configuring a remote administration server in the openFT manuals "openFT V11.0 for Unix Systems - Installation and Administration" and "openFT V11.0 for Windows Systems - Installation and Administration".

ADM administrator

Person who administers the remote administration server. This person creates the configuration data for remote administration in which, for instance, the remote administrators and the administered openFT instances are defined. The ADM administrator is the only person permitted to change the configuration data.

Remote administrator

Role configured on the remote administration server and which grants permission to execute certain administration functions on certain openFT instances. A remote administrator can

- Log in directly at the remote administration server (single sign-on)
- Log in to a different openFT instance (as of V11.0) and access the remote administration server using an FTAC transfer admission.
  The openFT instance can be running either on a mainframe (BS2000/OSD, z/OS) or on a Unix or Windows system. The FTADM protocol is used for communication.

Several remote administrators can be configured with different permissions.

Administered openFT instance

openFT instance that is able to be administered by remote administrators during live operation. Access is via an admission profile. The following applies, depending on the openFT version of the openFT instance:

- In the case of openFT instances as of V11.0, the FTADM protocol is used, and the full range of remote administration functions can be utilized.
- In the case of openFT instances from V8.0 through V10.0, administration is carried out using the openFT protocol and the command *ftexec*. The range of functions available depends on the openFT version of the instance being administered.

5.1.2 Configuring an openFT instance on z/OS for remote administration

The remote administration server uses FTAC transfer admissions to access the openFT instances. This means that the appropriate admission profiles must be defined in the openFT instances from which administration is being carried out.

To enable a remote administrator to access the openFT instance, the FT administrator sets up an admission profile on the z/OS system using the REMOTE-ADMINISTRATION function:

```
FTCREPRF NAME=profile                             -
         ,TRANSFER-ADMISSION=transfer admission      -
         ,PARTNER=remote administration server   -
         ,FT-FUNCTION=*REMOTE-ADMINISTRATION
```

The ADM administrator specifies the FTAC transfer admission in the configuration file of the remote administration server when defining the openFT instance. For an example, see the manual "openFT V11.0 for Unix Systems - Installation and Administration". The operand PARTNER= ensures that this profile can only be used by the remote administration server.

Entering the remote administration server in the partner list

If remote administration requests are to be issued from your z/OS system, the FT administrator can enter the remote administration server in the partner list. This has the advantage that you can explicitly assign particular attributes to this partner, for instance the security level or the trace settings.

The FT administrator enters the remote administration server in the partner list using the following format:

```
ftadm://host[:port number]
```

You only specify *port number* if the default ADM port (11000) is not used on the remote administration server *host*. The same applies if a remote administrator specifies the address directly in a remote administration request.

5.1.3 Issuing remote administration requests

If you wish to enter remote administration requests, you require the following:

- the name of the remote administration server in the partner list or the address of the remote administration server (ask the FT administrator if necessary)
- the transfer admission for accessing the remote administration server. The ADM administrator of the remote administration server must make this available to you.

You are able to determine the names of the openFT instances that you are permitted to administer yourself.

Determining the names of the openFT instances

The ADM administrator defines the names of the openFT instances during configuration of the remote administration server. You get the names of the openFT instances by executing the ftshwc command as a remote administration command on the remote administration server:

```
FTADM PARTNER-SERVER=server                           -
     ,TRANSFER-ADMISSION=transfer admission           -
     ,ROUTING-INFO=*NONE                              -
     ,CMD='ftshwc -rt=i'
```

Explanation

server

Name of the remote administration server from the partner list. Alternatively, you can also enter the address directly in the format *ftadm://host...*

transfer admission

FTAC transfer admission on the remote administration server.

'ftshwc -rt=i'

'ftshwc -rt=i' is a command executed on the remote administration server that outputs the names of the instances that you are permitted to administer. You must enter the quotes.

Sample output

```
TYPE    = *INSTANCE     ACCESS = FT+FTOP+FTAC
   NAME = Muenchen/Jonny
   DESC = Computer Test-en-lp
TYPE    = *INSTANCE     ACCESS = FTOP
   NAME = Muenchen/Hello
   DESC = Computer Hello
```

NAME specifies the name of the instance that you must specify exactly as given here in the remote administration request. Your remote administration permissions for this instance are listed under ACCESS. See also Abschnitt „Remote administration commands“ auf Seite 205.

Issuing a remote administration request

Specify the remote administration command in the following form:

```
FTADM PARTNER-SERVER=server                           -
     ,TRANSFER-ADMISSION=transfer admission           -
     ,ROUTING-INFO=instance                           -
     ,CMD='command'
```

Explanation

server

Name of the remote administration server from the partner list. Alternatively, you can also enter the address directly in the format *ftadm://host...*

transfer admission

FTAC transfer admission on the remote administration server.

instance

Routing name of the openFT instance on which the administration command is to be executed. You must enter this name in exactly the form in which it appears on the remote administration server with the ftshwc command. See „Determining the names of the openFT instances“ auf Seite 160.

command

Specifies the administration command to be executed on the openFT instance. For further details, see Abschnitt „FTADM Execute remote administration command“ auf Seite 202.

5.1.4 Logging remote administration

ADM log records are created in each of the openFT instances involved when remote administration requests are issued.

ADM log records are explicitly flagged as being of a particular type (A). They are handled in a similar way to FT or FTAC log records, i.e. you can view ADM log records in z/OS using the FTSHWLOG command (see Seite 311) and delete them with the FTDELLOG command (provided that you have the appropriate permission to do so, see Seite 234).

Controlling ADM logging

The FT administrator controls the scope of ADM logging using the operating parameters. The following options are available:

- log all administration requests
- log all administration requests that modify data
- log administration requests during which errors occurred
- disable ADM logging

You do this by means of the FTMODOPT command with the operand LOGGING= *SELECT(ADM=...)

5.2 ADM traps

ADM traps are short messages that openFT sends to the **ADM trap server** if certain events occur during operation of openFT. Such events may include errored FT requests, status changes or the unavailability of partners, for instance.

The ADM traps are stored permanently on the ADM trap server. This allows one or more openFT systems to be monitored at a central location. The FT administrator of the ADM trap server is thus provided with a simple way of gaining an overview of events that have occurred on the openFT instances he is monitoring.

If the ADM trap server is simultaneously used as a remote administration server, remote administrators can also view traps from other systems and hence monitor the systems that they are administering. This means that if you are a remote administrator, you can view the ADM traps of "your" administered instances on the z/OS.

5.2.1 Configuring ADM traps in the openFT instance

To allow ADM traps from your openFT instance on the z/OS system to be sent to the ATM trap server, you must carry out the following actions in your role as FT administrator:

- Enter the address and admission data for the ADM trap server
- Specify the scope of the ADM traps sent to the ADM trap server

In addition, the FT administrator of the ADM trap server must set up a corresponding admission profile on the ADM trap server.

Enter the address and admission data for the ADM trap server

You specify the address and the transfer admission of the ADM trap server in the ADM-TRAPS operand of the FTMODOPT command:

```
FTMODOPT ...                                               -
         ,ADM-TRAPS=*PAR(DESTINATION=(PARTNER=adm-trap-server,  -
                     TRANSFER-ADMISSION=trap-admission))
```

adm-trap-server
: must be defined in the partner list using the address format *ftadm://host...*. Alternatively, you can also enter the address directly in the format *ftadm://host...*

trap-admission
: is the transfer admission for the admission profile defined in the ADM trap server for this purpose.

Specify the scope of the ADM traps

The scope of the ADM traps sent to the ADM trap server is controlled using the operating parameters. You can set which of the events listed below cause traps to be sent:

- Change of openFT status (FTSTART / FTSTOP)
- Change of partner status
- Unavailability of partners
- Change of request management status
- Successfully completed requests
- Failed requests

To do this, use the FTMODOPT command and defying the required selection under SELECTION in the ADM-TRAPS operand.

5.2.2 Viewing ADM traps

The FT administrator of the ADM trap server is permitted to view all ADM traps on the ADM trap server. If the ADM trap server is also used as the remote administration server, the remote administrators can also view traps.

If you log on to your z/OS system as a remote administrator, you can view your "own" ADM traps. These are the ADM traps of those openFT instances for which you have at least FTOP permission. See the „Determining the names of the openFT instances“ auf Seite 160.

If you wish to view the most recent 10 ADM traps, enter the following remote administration command:

```
FTADM PARTNER-SERVER=server                          -
     ,TRANSFER-ADMISSION=transfer admission          -
     ,ROUTING-INFO=*NONE                             -
     ,CMD='ftshwatp -nb=10'
```

Explanation

server

Name of the remote administration server from the partner list. Alternatively, you can also enter the address directly in the format *ftadm://host...*

transfer admission

FTAC transfer admission on the remote administration server.

'ftshwatp -nb=10'

'ftshwatp -nb=10' is a command executed on the remote administration server that outputs the last 10 ADM traps. You must enter the quotes.

The ftshwatp command also provides further options. For details, see, for instance, the manual "openFT V11.0 for Unix Systems - Installation and Administration".

6 Command interface for the FT administrator

This chapter describes the commands you can use to administer openFT.

FT administrators are defined in the FTADM member of the openFT parameter library (PARM) while FTAC administrators are defined in the FTACADM member.

The openFT administration commands always apply to the currently set openFT instance. For this reason, you may need to assign the instance

– The CLIST <openft qualifier>.<inst>.CLIST must be concatenated
– The file <openft qualifier>.<inst>.CONN must be allocated with the DD name OPENFT. Exception: If the default instance is used exclusively and the OPFT subsystem uses extended code 211. In this case also, it is urgently recommended that the file <openft qualifier>.<inst>.CONN is allocated.

The specifications for OPENFT QUALIFIER and the INSTANCE NAME are defined when the instance is set up with FJGEN.

All administration commands except FJGEN can be entered directly at a TSO dialog terminal. The FT administration mode (FJADM) is **no longer** supported in the current version. For an assignment of the old FJADM program statements to the more recent TSO commands, refer to the Appendix (page 389).

For information on how the FT administration commands can be used in a z/OS system without TSO, consult the Appendix page 488 ff.

6.1 Functional command overview

The following overview shows the FT and FTAC administrator commands as they relate to individual jobs.

6.1.1 FT command overview

Set, output or update installation parameters

Setting installation parameters	FJGEN	page 182
Output installation parameters	FJGENPAR	page 191
Update operating parameters	FTUPDPAR	page 365

Administer partners

Add partner to the partner list	FTADDPTN	page 195
Remove partner from the partner list	FTREMPTN	page 303
Modify partner properties	FTMODPTN	page 295
Display partner systems	FTSHWPTN	page 352

Load, activate, and deactivate or terminate openFT

Load openFT	FJINIT	page 193
Activate openFT	FTSTART	page 361
Deactivate openFT	FTSTOP	page 362
Terminate openFT	FTTERM	page 363

Controlling openFT operating parameters

Modify operating parameters	FTMODOPT	page 254

Administer key pair sets for authentication

Create a key pair set	FTCREKEY	page 210
Update public keys	FTUPDKEY	page 364
Delete a key pair set	FTDELKEY	page 233

Remote administration

Issue remote administration command	FTADM	page 202

Manage request queue		
Cancel FT requests	NCANCEL, also FTCANREQ	page 366
Show information on FT requests	NSTATUS, also FTSHWREQ	page 371
Modify FT request queue	FTMODREQ	page 300
Logging Function		
Delete logging records	FTDELLOG	page 234
Show logging records	FTSHWLOG	page 311
Display information on reason codes in the logging records	FTHELP	page 243
Monitoring		
Show monitoring data	FTSHWMON	page 327
Obtain information on openFT		
Display the network environment	FTSHWNET	page 338

6.1.2 FTAC commands overview

openFT-AC must be installed in order to use the following commands:

Edit FTAC admission profiles

Create admission profile	FTCREPRF	page 212
Delete admission profile	FTDELPRF	page 238
Modify admission profile	FTMODPRF	page 273
Show admission profile	FTSHWPRF	page 346

Edit FTAC admission sets

Modify admission set	FTMODADS	page 248
Show admission set	FTSHWADS	page 304

Store and display saved FTAC admission profiles and sets

Export admission profiles and sets	FTEXPENV	page 241
Import admission profiles and sets	FTIMPENV	page 245
Display saved admission profiles and sets	FTSHWENV	page 308

Show partner systems

Display partner systems and security levels	FTSHWRGE	page 358

6.2 Entering FT commands

Please remember the following when entering commands:

- You must insert commas to separate the individual operands of a command, e.g.

  ```
  NCOPY TRANSFER-DIRECTION=TO,PARTNER-NAME=ZENTRALE,LOCAL-PARAMETER =...
  ```

- If quotes appear in a value assignment which is itself enclosed in quotes, they must be entered twice.
- If there is no default value marked (by underscoring) for an operand, then it **must** be specified with a valid value (mandatory operand).
- A distinction is made between positional operands and keyword operands. Positional operands are uniquely determined by their position in the command while keyword operands are uniquely determined by their keyword, for example `TRANSFER-DIRECTION=...` There are a number of considerations to be borne in mind when specifying such operands (see below).
- You can abbreviate your entries for commands and operands, always ensuring that your entries retain their uniqueness. You can also use positional operands if you wish. Short forms and long forms can be mixed at will. Certain abbreviated forms of keywords and a number of positional operands are guaranteed for openFT. This means that you will find these options unchanged in subsequent versions. This means, therefore, that to be “on the safe side”, you should form the habit of entering these commands in their abbreviated form. You should take particular care to use the guaranteed abbreviated forms in procedures, as this will ensure their continued executability in subsequent versions. The recommended abbreviations are used in the examples shown in this chapter, and the possible abbreviations are listed for the individual command formats.
- If a structure is preceded by an introductory operand value (e.g. *BS2000 is an introductory operand value in REM=*BS2000(...)), then the opening parentheses must immediately follow this operand value. Introductory operand values may be omitted if there is no risk of ambiguity.
- The asterisk (*) that precedes constant operand values may be omitted if there is no risk of ambiguity. Please ensure that it is not a guaranteed abbreviation.
- Comments may be included in FT user commands using the form "..."; the normal method of including comments in other TSO commands using the form /*...*/ is not permitted.

When you enter commands, the value assignments for the operands may be specified in positional form, in keyword form or in mixed form.
Please note the following:

– When you perform value assignments in positional form, the first value is assigned to the first operand in the command, the second value to the second operand etc.

– Values assigned in positional form are separated by commas. You must also enter a comma for each operand for which no value is assigned.

– If two values are assigned to an operand, the last value to be assigned always applies. This also applies to parameter specifications in introductory operand values within the corresponding structure brackets. However, for the sake of clarity, double assignments should generally be avoided.

– If you mix the different forms of operand value assignments (positional and keyword form), then you must observe the correct sequence. Note that you can start your input with positional operands and follow these with keyword operands but not the other way round!

– Since there is a possibility that the sequence of operands may change in subsequent versions, only keyword operands should be used in procedures.

Continuation lines in FT commands in z/OS

An NCOPY command may consist of more than one line. When entering an NCOPY command with continuation lines at a TSO terminal, you simply continue writing on the next line on the screen.
If an NCOPY command with continuation lines is issued in a CLIST or REXX procedure or in a batch job as data for the IBM utility IKJEFT01, a hyphen "-" or a plus sign "+" is used as the continuation character. Refer to the IBM manuals for more details.

Differentiation between uppercase and lowercase letters

It may be important to differentiate between uppercase and lowercase letters in the parameters.

openFT handles the letters contained in the command string according to the following rules:

1. If the command string received by openFT contains only uppercase letters,
   - all letters outside the quotation marks remain uppercase;
   - letters enclosed in quotation marks are converted to lowercase.
   - alphanumerically specified FTAC transfer admissions are converted into lowercase letters

2. If any part of the command string received by openFT except the command name (NCOPY) contains a lowercase letter,

   – all letters outside the quotation marks are converted to uppercase;

   – alphanumerically specified FTAC transfer admissions are converted into lowercase letters

   – letters enclosed in quotation marks are not converted. These letters are retained in the form in which they were entered.

This has the following consequences for command input:

If parameter values consisting of uppercase letters (or of both uppercase and lowercase letters) enclosed in quotation marks are to be entered, you must ensure that

- the command contains at least one lowercase letter (at any position except in the command name) and
- openFT receives this command string in the same form (with no conversion).

This means that

- In a CLIST or REXX procedure, you must use the statement CONTROL ASIS (or CONTROL NOCAPS) to ensure that the command string is not converted to uppercase before execution.
- You can also use the menu interface (see page 143); here, the relevant fields are not converted to uppercase (see the description of the input fields in the data entry panels).
- When the TSO command processor is called in a batch job (IBM utility IKJEFT01, see section "Using openFT in z/OS systems without the TSO interactive system" on page 488), letters are not converted to uppercase.

These rules also apply to the hexadecimal digits A through F in entries of the form <x-string m..n> which expect the partner system to be specified in uppercase letters.

6.3 Command syntax representation

The command format consists of a field with the command name. All operands with their legal values are then listed. Operand values which introduce structures and the operands dependent on these operands are listed separately. The syntax of the command representation is explained in the following three tables.

table 1: Notational conventions

The meanings of the special characters and the notation used to describe command and statement formats are explained in table 1.

table 2: Data types

Variable operand values are represented in SDF by data types. Each data type represents a specific set of values. The number of data types is limited to those described in table 2.

The description of the data types is valid for the entire set of commands/statements. Therefore only deviations (if any) from the attributes described here are explained in the relevant operand descriptions.

table 3: Suffixes for data types

Data type suffixes define additional rules for data type input. They contain a length or interval specification.

The description of the data type suffixes is valid for the entire set of commands/statements. Therefore only deviations (if any) from the attributes described here are explained in the relevant operand descriptions.

Metasyntax

Representation	Meaning	Examples
UPPERCASE LETTERS	Uppercase letters denote keywords (command, statement or operand names, keyword values) and constant operand values. Keyword values begin with *	HELP-SDF SCREEN-STEPS = *NO
UPPERCASE LETTERS in boldface	Uppercase letters printed in boldface denote guaranteed or suggested abbreviations of keywords.	GUIDANCE-MODE = *YES
=	The equals sign connects an operand name with the associated operand values.	GUIDANCE-MODE = *NO
< >	Angle brackets denote variables whose range of values is described by data types and suffixes (see Tables 2 and 3).	SYNTAX-FILE = <filename 1..54>
Underscoring	Underscoring denotes the default value of an operand.	GUIDANCE-MODE = *NO
/	A slash serves to separate alternative operand values.	NEXT-FIELD = *NO / *YES
(...)	Parentheses denote operand values that initiate a structure.	,UNGUIDED-DIALOG = *YES(...) / *NO
[ ]	Square brackets denote operand values which introduce a structure and are optional. The subsequent structure can be specified without the initiating operand value.	SELECT = [*BY-ATTRIBUTES](...)

Indentation	Indentation indicates that the operand is dependent on a higher-ranking operand.	,GUIDED-DIALOG = *YES(...)
		*YES(...)
		SCREEN-STEPS = *NO /
		*YES

Table 1: Metasyntax (part 1 of 2)

Representation	Meaning	Examples
|	A vertical bar identifies related operands within a structure. Its length marks the beginning and end of a structure. A structure may contain further structures. The number of vertical bars preceding an operand corresponds to the depth of the structure.	**SUP**PORT = ***TAPE**(...) ***TAPE(...)** | **VOL**UME = ***ANY**(...) | ***ANY**(...) | | ...
,	A comma precedes further operands at the same structure level.	**GUID**ANCE-**MODE** = ***NO** / ***Y**ES ,**SDF-COM**MANDS = ***NO** / ***Y**ES
list-poss(n):	The entry “list-poss” signifies that a list of operand values can be given at this point. If (n) is present, it means that the list must not have more than n elements. A list of more than one element must be enclosed in parentheses.	list-poss: ***SAM** / ***ISAM** list-poss(40): <structured-name 1..30> list-poss(256): ***OMF** / ***SYSLST**(...) / <filename 1..54>
Alias:	The name that follows represents a guaranteed alias (abbreviation) for the command or statement name.	**HELP-SDF** Alias: **HPSDF**

Table 1: Metasyntax (part 2 of 2)

Data types

Data type	Character set	Special rules
alphanum-name	A…Z 0…9 $, #, @	
c-string	EBCDIC character	Must be enclosed within single quotes; the letter C may be prefixed; in the case of file names in z/OS it must be prefixed; any single quotes occurring within the string must be entered twice.
composed-name	A…Z 0…9 $, #, @ Hyphen Period	Alphanumerical string that can be subdivided into multiple substrings by periods or hyphens.
date	0…9 Structure identifier: hyphen	Input format: yyyy-mm-dd yyyy: year; optionally 2 or 4 digits mm:month dd:day Only date specifications between 1.1.2000 and 19.1.2038 are possible. If the year is specified in 2-digit form, 2000 is added to the number
filename	A…Z 0…9 $, #, @ hyphen period Colon Single quote	Input format fully qualified: ':<prefix>:<first-qual>.<filename>' Input format partially qualified: :<prefix>:<filename> :<prefix>: Optional specification of file organization; enclosed in colons; can assume the following values: :S: for PS :O: for PO :E: for PDSE :L: for PO or PDSE :V: for VSAM

Table 2: Data types (part 1 of 3)

Data type	Character set	Special rules
		<first-qual> "first level qualifier" User ID (max. 7 characters, character range A…Z, 0…9, $, #, @; may not begin with a digit) or alias (max. 8 characters) <filename> partially qualified file name; the syntax of z/OS file names depends on the file organization; refer to the overview in the user manual "openFT for and z/OS - Managed File Transfer in the Open World"
filename-prefix	A…Z 0…9 $, #, @ hyphen period Colon Single quote	Input format fully qualified: ':<prefix>:<first-qual>.<partname>.' or ':<prefix>:<first-qual>.<partname>/' Input format partially qualified: :<prefix>:<partname>. or :<prefix>:<partname>/ <prefix> see filename <first-qual> see filename partname Specifies the common first part of the partially qualified name of files. partname must be followed by a period or a slash.
integer	0…9, +, -	+ or -, if specified, must be the first character.
name	A…Z 0…9 $, #, @	Must not begin with 0...9.
number	0...9 A...F	Message number/return code

Table 2: Data types (part 2 of 3)

Data type	**Character set**	**Special rules**
partial-filename	A…Z 0…9 $, #, @ hyphen period	Input format fully qualified: ':<prefix>:<first-qual>.<partname>.' Input format partially qualified: :<prefix>:<partname>. <prefix> see filename <first-qual> see filename partname Specifies the common first part of the partially qualified name of files. partname must be followed by a period.
text	freely selectable	For the input format, see the relevant operand descriptions.
time	0…9 structure identifier: colon	Time-of-day entry: Input format: { hh:mm:ss hh:mm hh } hh:hours mm:minutes ss:seconds } Leading zeros may be omitted Valid entries are between 00:00:00 and 23:59:59.
x-string	Hexadecimal: 00…FF	Must be enclosed in single quotes; must be prefixed by the letter X. There may be an odd number of characters.

Table 2: Data types (part 3 of 3)

Suffixes for data types

Suffix	Meaning
x..y	With data type "integer": interval specification x minimum value permitted for "integer". x is an (optionally signed) integer. y maximum value permitted for "integer". y is an (optionally signed) integer.
x..y	With the other data types: length specification For data types date and time the length specification is not displayed. x minimum length for the operand value; x is an integer. y maximum length for the operand value; y is an integer. x=y the length of the operand value must be precisely x.

Table 3: Suffixes for data types

Meaning of operands

After the format of each command there is a detailed description of all the operands, the possible value assignments and their functions.

Otherwise the same metasyntax is used in describing operands as in the representation of the command formats (see above).

The following characters are regarded as constants in describing the operands: "." (period), "(" (open bracket), ")" (close bracket), " ' " (single quote), "$" (dollar sign), and also the character combinations ":V:", ":L:", ":S:", ":O:" and ":E:" i.e. they must be specified when the command is entered. Where this occurs the syntactical components of the operand value must follow one after another without any gaps.
"±" has the usual meaning "+" or "-".

Example

Possible entries for the local operand FILE are as follows:

```
ABC                   'USER1.ABC',                (1)
GROUP1.G1234V01       'USER1.GROUP1.G1234V01'     (2)
GROUP2(+27)           'USER1.GROUP2(+27)'         (3)
GROUP3(0)             'USER1.GROUP3(0)'           (4)
:V:VSDAT              ':V:USER1.VSDAT'            (5)
PDS1(DEF)             'USER1.PDS1(DEF)',          (6)
:L:PODS2              ':L:USER1.PODS2'            (7)
./directory5/abcd     /u/user002/directory5/abcd  (8)
```

Key

(1) Name of a PS data set

(2) Name of an absolute generation data set (PS data set) (this has the same syntax as the name of a normal PS data set, with the exception of the last partial name, which must have a special format)

(3) Name of a relative generation data set (PS data set)

(4) Name of a relative generation data set (PS data set), special case "current generation" (may only be a send file)

(5) Name of a VSAM file of the type "entry sequenced"

(6) Name of a PO or PDSE member

(7) Name of an entire PO or PDSE data set

(8) Pathname of an openEdition file (absolute and relative)

More details on the syntax rules for file names, passwords, user IDs and account numbers in openFT can be found in the respective sections in chapter 3 of the User Guide.

6.4 Command return codes

The TSO commands supply a return code that provides information about whether command processing has succeeded or failed. It is stored in the TSO's system variable ("control variable") &LASTCC. A return code other than 0 is generated only if a corresponding message is output at the terminal. These messages are described in the Appendix (page 430ff).

This return code may have the following values:

`Return-Code = 0:`

The command was accepted. (Corresponds, for example, to the message FTR0000 or FTR0008 at the terminal.)

`Return-Code = 4:`

The command was accepted with a minor warning, for example if no corresponding administration objects were found.

`Return-Code = 8:`

Reserved

`Return-Code = 12 (or > 12):`

The command was rejected due to an error. The request was not accepted.

The TSO commands can also be started in response to an ftexec command that was started in a remote Unix or Windows partner system. The partner system is sent either the return code 0 (if the command was accepted) or 12 (if the command was terminated with an error).

6.5 Output in CSV format

The output of some SHOW commands in openFT and openFT-AC can be optionally requested in CSV (Comma Separated Values) format. CSV is a popular format in the PC environment in which tabular data is defined by lines. Output in CSV format is offered for the following commands:

- NSTATUS
- FTSHWENV
- FTSHW *
- FTSHWADS
- FTSHWLOG
- FTSHWMON
- FTSHWOPT
- FTSHWPTN
- FTSHWPRF
- FTSHWRGE

  * see User Guide

Many programs such as spreadsheets, databases, etc., can import data in CSV format. This means that you can use the processing and presentation features of such programs on the data output by the above commands.

The output fields are described in the appendix.

The first line is the header and contains the field names of the respective columns. **Only the field names are guaranteed, not the order of fields in a record.** In other words, the order of columns is determined by the order of the field names in the header line.

6.6 FJGEN Set installation parameters

Note on usage

User group: FT administrator

Functional description

You use the TSO procedure FJGEN to set up a new openFT instance or to modify the parameter settings of existing instances. The FJGEN command can only be issued in TSO command mode:

```
EXEC <FT-basic-procedure-library>(FJGEN)
```

where <FT-basic-procedure-library> must be replaced by the CLIST present in the FJGEN command (generally OPENFT.CLIST under the openFT installation ID).

FJGEN starts a dialog that requests the installation parameters for the openFT instance. Filenames must be entered with the user ID but without single quotes.

FJGEN uses the installation parameters to create installation-specific CLISTs and the JCL for an installation-specific batch job (see below). These procedures are required for the administration of openFT. FJGEN stores them in the FT procedure library (CLIST library):

```
<openft qualifier>.<inst>.CLIST
```

The first two name parts here are replaced by OPENFT QUALIFIER and INSTANCE NAME.

FJGEN can also be used without an operand to modify the installation parameters; the procedures mentioned are then regenerated. The changes take effect the next time the installation-specific batch job is started with FJINIT.

The batch job and the FJINIT command are located in the CLIST generated by FJGEN: <openft qualifier>.<inst>.CLIST.

Even if openFT is running as a started task, the installation parameters are modified with FJGEN. The FT administrator must make the necessary changes in the start procedure himself. See section “openFT as a job or started task” on page 85. In the case of parameters that are queried by FJGEN but are not required for the started task, the best solution is to enter an "x" in FJGEN.

Note

You can also store some of the installation parameters in the PARM member of the FT parameter library and pass them to openFT; specifications of this kind overwrite the specifications made for FJGEN. Further information is given in the section “Setting up the FT parameter library” on page 49.

Example: Set installation parameters (FJGEN without an operand)

```
fjgen
 ******  FJGEN/V110A00 INSTALLATION PROCEDURE openFT V11.0A00 ******
 ENTER INSTANCE NAME       : (DEFAULT: STD)
 ENTER FT-LOADLIB          : USERA.openft.load
 ENTER FT-NCLOADLIB        : USERA.openft.NCLOAD
 ENTER VOLUME/UNIT         : vsn123/sysda
 ENTER openFT USER ID      : openft
 ENTER openFT USER ACCOUNT : (a123,b123)
 ENTER openFT USER PASSWORD: openft
 ENTER OPENFT QUALIFIER    : openftqu
 ENTER FT-ID               : ftid1
 ENTER FT-PASSWORD         : affe
 ENTER RUNMODE             : S(TANDARD)/A(UTOMATIC) a
 ENTER FT-PARMLIB          : openftqu.std.parm
 ENTER CMDPORT             : 1100
 ENTER HOST NAME           :
 ENTER HSM-MCDS NAME       :

 FJGENPAR CREATED
 FJINIT   CREATED
 FJBATCH  CREATED
 FJVERS   CREATED (FUNCTION: GET VERSION OF LOADMODULS)
 ******   FJGEN END   ******
 READY
```

i The FT procedure library FT-PROCLIB is additionally displayed on the subsequent call to FJGEN 'INFO' or FJGENPAR. See page 184.

The various items of information requested or displayed have the following meaning:

INSTANCE NAME
The instance name is used to administer the openFT instance. It may be up to 5 characters in length. If this entry is omitted then the instance name STD is set.
The names of instances that are to be switched within a computer cluster must be unique within the cluster.

The instance name identifies the components that belong to an openFT instance (data sets) and is used to address these internally (see also section “Setting up openFT instances” on page 136). However, it should not be confused with the instance identifier (this is defined for the purposes of address information with the FTMODOPT command).

FT-PROCLIB
Name of the FT procedure library (CLIST library). This is only displayed with FJGEN ’INFO’ or FJGENPAR (see page 191). If it does not already exist, this PO file is created automatically when FJGEN is called. FJGEN stores the command procedures for the openFT instance in this library. It has the fixed name
<openft qualifier>`.<inst>.CLIST`
The first two name parts here are replaced by OPENFT QUALIFIER and the instance name.

FT-LOADLIB
Name of the FT load module library. This PO or PDSE data set must contain the following load modules: OPENFT, OPENFTSL and OPFTSUBL. The name of the library must be entered including its user ID but without single quotes.

FT-NCLOADLIB
Name of the FT load module library for openFT commands such as FTSHWPTN, NCOPY, etc. Among other things, this PO or PDSE file must contain the load modules FTATTP and FTDETP. The name of this library must be entered with the user ID but without quotes.

VOLUME/UNIT
VSN (volume serial number) and group name (unit) of the disk containing the request file, the partner list, the log file, the FTAC file, the trace files and the dump files, if any (see section “Internal openFT data sets” on page 424).

If the corresponding files are SMS managed, the specifications for VOLUME and UNIT may have no effect under certain circumstances. If the files are not SMS managed, an "SMS managed volume" must not be specified here.

You can specify both values; if you only want to specify one of the two values, it may be necessary to use a slash to distinguish which value you want to specify. If you want to specify neither VOLUME nor UNIT (i.e. only a slash), openFT assumes the UNIT name DASD. This UNIT name must therefore be defined in the system. (You can also define the volume for the request file, the partner list and the volume for the trace files and dump files via the corresponding parameters in the PARM member. Specifications in PARM overwrite the specifications made for FJGEN. Further information is provided in the section “Setting up the FT parameter library” on page 49. The assumption is made here that no volume specifications are made in PARM.)

Examples

VOLUME/UNIT	VOLUME	UNIT
VSN123/SYSDA	VSN123	SYSDA
VSN123 or VSN123/	VSN123	---
/SYSDA	---	SYSDA
/	---	DASD

openFT USER ID
User ID under which the openFT job is to execute. Once an instance has been set up, this user ID is also authorized by default to administer FT and possibly also FTAC.

openFT USER ACCOUNT
Accounting information for the job under which openFT is to execute. If the accounting information contains more than one parameter it must be specified in parentheses (see IBM manual "MVS/ESA JES2 Commands").

Null input is permissible if no accounting information is required.
Maximum length of accounting information: 40 characters.

openFT USER PASSWORD
Password for the user ID under which openFT is to execute.

OPENFT QUALIFIER
Qualifier for the instance-specific files. The OPENFT QUALIFIER may be up to 17 characters in length and may contain maximal a period. Hence, It may consist solely of a "first level qualifier" or a "first level qualifier" and a "second level qualifier".

Please note the following:

- The "second level qualifier" in the OPENFT QUALIFIER may consist of at most one character if ADM traps are to be output.
- Trace file names can be shortened if the OPENFT QUALIFIER contains a "second level qualifier".

FT-ID
FT identifier. This character string can consist of up to 5 alphanumeric characters must be unique among all FT systems interconnected via a SNA network. If an SNA network is not used either for internal communication or for interconnections with other FT systems, then you can specify any value for FT-ID (preferably an 'x').

FT-PASSWORD
FT password. This password serves to protect the VTAM applications, the request file, the partner list and the trace files.

This parameter must be specified even if these resources are not password-protected.

RUNMODE
specifies the openFT start mode:

S or SS the FJINIT command merely loads openFT.

A or AA the FJINIT command loads and immediately activates openFT (the FJSTART command is superfluous in this case).

**D snap dumps can be generated for diagnostic purposes.

"**" stands for "SS" or "AA" with the same meaning as above.

openFT can only be loaded in non-privileged mode for test purposes.

⚠ **WARNING!**
In non-privileged mode, openFT does not check the transfer admission or data access authorization. This means that:

– Transfer requests are accepted and executed even if invalid specifications are made in the TRANSFER-ADMISSION or PROCESSING-ADMISSION.

– The execution of other functions, e.g. follow-up processing or the printing of result lists, however, can be rejected by the system if invalid specifications are detected which openFT has not rejected.

openFT is loaded with the following specifications in non-privileged mode:

N or NS openFT is loaded in non-privileged mode.

NA openFT is loaded in non-privileged mode and activated immediately.

NSD openFT is loaded in non-privileged mode. Diagnostic capabilities are activated.

NAD As above.

FT-PARMLIB
Name of the openFT parameter library. If no name is entered for this library in FJGEN then openFT uses the default value:
<openft qualifier>.<inst>.PARM
The first two name parts are replaced by OPENFT QUALIFIER and the name of the instance.

If the parameter library does not exist at the time FJGEN is called, openFT creates it with the following content:

– PARM member with the entries:

```
CMD_TRANS=TCP
DSTYPEDEF=PS
LIBTYPEDEF=PO
OPENFT_SVC=211
```

  The entry OPENFT_SVC is important if openFT is to perform command encryption using the started openFT subsystem. See section “Providing the OPFT subsystem” on page 85. If the openFT subsystem is not available or not started, the openFT batch job or the started task can only be started if this entry is deleted or invalidated.

– TNSTCPIP member with the DUMMY entry

```
SAMPLE=255.255.255.255:1100:$FJAM   : SAMPLE ENTRY
```

– FTADM with the entries

```
"OPENFT USER ID" from FJGEN
Console
```

– FTACADM with the entries

```
"OPENFT USER ID" from FJGEN
Console
```

Details on the parameter library and its members can be found in the section “Setting up the FT parameter library” on section “Setting up the FT parameter library” on page 49).

CMDPORT
Port number of the command client, i.e. the port number of the current openFT instance for connecting the interactive tasks to openFT. CMDPORT is only relevant if the CMD TRANS parameter is not set to VTAM in the PARM member of the openFT parameter library. If no port number is specified here, openFT uses the openFT-specific default port number 1100.

HOST NAME
Host name for the current openFT instance. This information is required for addressing in TCP. The host should be specified directly as an IP address or as a hostname. If a member with the name TNSTCPIP still exists in the library PARM, and if this member is to be used, it is also possible to specify a name of up to 8 characters that refers to an entry in TNSTCPIP (where it is converted into an IP address). If multiple openFT instances are to be able to run in parallel with TCP/IP then they must be assigned different IP addresses. Please note that you may only use IP addresses that are defined in your z/OS system’s address space. If you do not specify this value, openFT uses the first IP address that is defined in the z/OS system.

HSM-MCDS NAME
Help file for archiving and restoring (migrating) files. If nothing is specified, openFT sets the default value DFHSM.MCDS.

FJGEN uses the specified installation parameters, for example, to create the following JCL statements for a batch job for loading and starting the openFT load module (these statements are stored in the FJBATCH member of the FT procedure library):

```
//OPENFTF   JOB (A123,B123),                                       (1)
//            CLASS=A,MSGCLASS=A,
//            USER=OPENFT,PASSWORD=OPENFT,                         (2)
//            TIME=1440,REGION=0M
//DLTDMP   EXEC PGM=IEFBR14                                        (7a)
//DELFILE  DD OPENFTQU.STD.SYSMDUMP,                               (7a)
//            DISP=(MOD,DELETE,DELETE),
//            SPACE=(CYL,(20,5)),
//            DCB=(DSORG=PS)
//OPENFT   EXEC PGM=OPENFT,TIME=1440,
//            PARM='OPENFTQU,VSN123/SYSDA,A,FTID1,STD,AFFE,1100,'  (3)
//*           openFT V11.0A00 / FJBATCH V110A00
//STEPLIB  DD DSNAME=USERA.OPENFT.LOAD,                            (4)
//            DISP=(SHR,KEEP)
//OPENFTS  DD DSNAME=USERA.OPENFT.NCLOAD,                          (4a)
//            DISP=(SHR,KEEP)
//OPENFT   DD DSNAME=OPENFTQU.STD.CONN,                            (8)
//            DISP=(SHR,KEEP)
//OPFTATT  DD DSNAME=OPENFTQU.STD.OPFTATT,
//            DISP=(SHR,KEEP)
//*DDUADS   DD DSNAME=SYS1.UADS,                                   (5)
//*             DISP=(SHR,KEEP)
//OPFTHSM  DD DSNAME=OPENFTQU.STD.COLLECT.DATA,
//            DISP=(SHR,KEEP)
//MCDS     DD DSNAME=DFHSM.MCDS,DISP=SHR
//SYSIN    DD DUMMY
//SYSOUT   DD DUMMY
//IEBCOUT  DD DUMMY                                                (6)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD DSN=OPENFTQU.STD.SYSUDUMP,                           (7)
//         SPACE=(CYL,(20,5)),DISP=(,CATLG),
//         DCB=(DSORG=PS)
```

Explanation:

(1) Jobname = openFT USER ID with appended F "accounting information" as specified for openFT USER ACCOUNT. If multiple openFT instances are to run on your system, then different letters must be appended to the job names.

(2) openFT USER ID and openFT USER PASSWORD

(3) Parameter string consisting of the following start parameters:

Start parameter	Corresponding keywords in PARM
OPENFT QUALIFIER	No corresponding keyword
VOLUME/UNIT	NABVOLUME, NABUNIT, DMP_VOLUME, DMP_UNIT
RUNMODE	RUN_MODE[1]
FT-ID	FJAM_ID [1]
INSTANCE NAME	No corresponding keyword
FT-PASSWORD	FJAM_PASSW[1]
PORT	No corresponding keyword
HOST NAME	No corresponding keyword

[1] These keywords in PARM are only supported for reasons of compatibility.

The start parameters are ignored if you specify the corresponding parameters in FTMSPPAR. See also the section “Setting up the FT parameter library” on page 49

(4) FT-LOADLIB

(4a) FT-NCLOADLIB

(5) This DD statement is created as comment if the SYS1.UADS file exists on your system. If the batch job (or the start procedure for the started task) does not contain a DD statement of this type, openFT cannot check the passwords and account numbers of user IDs via SYS1.UADS (see the section “Linking openFT with data protection products” on page 90). In this case, the following message appears in the job log

```
IEC130I DDUADS DD STATEMENT MISSING
```

If user IDs are to be checked using SYS1.UADS, this DD statement must be activated by removing the comment asterisks.

If user IDs are to be checked using RACF and if a file SYS1.UADS is still present on the system, the comment character must **not** be removed.

If the batch job does contain a DD statement of this type, but the SYS1.UADS file does not exist on your system (i.e. was deleted from the system after the batch job was created using FJGEN), the batch job is terminated with a JCL error and the following message appears in the job log:

```
IEF212I ........ ...... DDUADS - DATA SET NOT FOUND
```

(6) openFT requires this DD statement in order to call the IBM utility IEBCOPY for transferring an entire PO or PDSE data set

(7) DD statement for the generation of a machine-readable dump in the case of system errors; the filename is preceded by the OPENFT QUALIFIER.

(7a) Statements for deleting the dump before the next openFT run.

(8) The instance-specific connection file

You can adapt the JCL statements created by FJGEN to the requirements of your own system. Modifications which affect the function of openFT are described in the section “openFT as a job or started task” on page 85.

6.7 FJGENPAR Output installation parameters

Note on usage

User group: FT administrator

This command must be called under TSO.

Functional description

You use the FJGENPAR command to output the openFT installation parameters on screen.

Format

FJGENPAR

Without operands

FJGENPAR outputs the installation parameters of the corresponding openFT instance as defined using FJGEN to the screen.

FJGENPAR only displays the original settings made during the FJGEN run.

Subsequent changes to the settings are not visible here, for instance:

- subsequent editing of the FJBATCH procedure or
- "overwriting" by corresponding parameters in the PARM member of the FT parameter library (such as RUN_MODE). See also page 189.

Example: outputting installation parameters

```
fjgenpar
INSTANCE NAME     : STD
FT-PROCLIB        : OPENFTQU.STD.CLIST
FT-LOADLIB        : USERA.OPENFT.LOAD
FT-NCLOADLIB      : USERA.OPENFT.NCLOAD
VOLUME/UNIT       : VSN123/SYSDA
openFT USER ID    : OPENFT
OPENFT QUALIFIER  : OPENFTQU
FT-ID             : FTID1
FT-PASSWORD       : AFFE
RUNMODE           : A
FT-PARMLIB        : OPENFTQU.STD.PARM
CMDPORT           : 1100
HOST NAME         :
HSM MCDS          : DFHSM.MCDS
READY
```

For the meaning of the output information, see the example accompanying the description of the FJGEN command (page 182).

6.8 FJINIT Load openFT

Note on usage

User group: FT administrator

This command can be entered in the TSO command mode only.

Functional description

You use the FJINIT command to load and start the openFT load module if openFT is to run as a background process

For information on loading and starting openFT as a started task, see the section “openFT as a job or started task” on page 85.

FJINIT

Without operands

Successful loading of openFT is acknowledged with the following message:

```
JOB useridF (JOBnnnnn) SUBMITTED.
```

The following message is output into the job logging file:

```
FTR4120 OPENFT: INITIATED
```

Notes

- The FJINIT command starts the member FJBATCH of the FT procedure library (see the description of the FJGEN command starting on page 182) defined at installation as a batch job by means of SUBMIT. The job name consists of the OPENFT USERID specified in the FJGEN command plus the letter "F". For technical reasons, the last letter of 8-character user IDs is replaced by an "F". If multiple openFT instances are to run in parallel under a user ID then the job names must end with different last letters. In this case, after running FJGEN, you should replace the "F" in the batch job with another letter (except for L,N, J, Z and P).
- No check is carried out as to whether openFT has already been loaded. If FJINIT is entered twice, the second job is delayed by the job scheduler until the first job has terminated.

– Depending on the openFT start mode, the local openFT instance can also be activated immediately when the openFT load module is loaded and started. In this case, it is not necessary to issue the FTSTART command.

  (See also the RUNMODE parameter in the description of the FJGEN command, page 182.)

6.9 FTADDPTN Add remote system to the partner list

Note on usage

User group: FT administrator

You can issue the FTADDPTN command under TSO.

Functional description

The FTADDPTN is used to enter a remote system in the partner list of the local openFT instance. The network or transport system must be generated beforehand.
For details concerning the generation process, please refer to the chapter “Installation and initial operation” on page 19 or another relevant manual.

The specifications which you need to enter for each partner system depend on the type of partner system (openFT for z/OS, openFT for BS2000, openFT for Unix systems, openFT for Windows etc.), and the method of connection to the remote system (SNA, TCP/IP directly). For details on specifying partner addresses, refer to section “Defining partner properties” on page 108.

If dynamic partners are permitted then inbound and outbound requests can be processed with partners which are accessed via their addresses and are not defined in the partner list.

You can issue the ADD-FT-PARTNERFTADDPTN command for all partner types while the FT system is running (openFT partners, ftp partners and ADM partner).

You can modify the partner system entry with FTMODPTN (page 295) and delete it with FTREMPTN (page 303).

Format

```
FTADDPTN

 PARTNER-NAME = <name 1..8> / *NONE
,PARTNER-ADDRESS = <text 1..200 with-low>
,SECURITY-LEVEL = *STD / *BY-PARTNER-ATTRIBUTES / <integer 1..100>
,STATE = *ACTIVE(...) / *DEACT
    *ACTIVE(...)
     |    AUTOMATIC-DEACT = *NO / *YES
,IDENTIFICATION = *STD / <composed-name 1..64> / <c-string 1..64 with-low>
,SESSION-ROUTING-INFO = *NONE / *IDENTIFICATION / <alphanum-name 1..8>
,PARTNER-CHECK = *BY-FT-OPTIONS / *STD / *TRANSPORT-ADDRESS
,TRACE = *BY-FT-OPTIONS / *ON / *OFF
,AUTH-MANDATORY= *NO / *YES
,PRIORITY= *NORMAL / *LOW / *HIGH
```

Operands

PARTNER-NAME =
Symbolic name of the partner system. It can be freely selected and need only be unique within openFT.

PARTNER-NAME = <name 1..8>
The operand value “name” consists of a maximum of 8 alphanumeric characters and must be unique in the local system. The FT administrator defines this name. This name can be used in the PARTNER parameter in all FT commands in order to address the partner system.

PARTNER-NAME = *NONE
Specifies that the partner is a dynamic partner.

PARTNER-ADDRESS = <text 1..200 with-low>
Address of the partner system. This specifies whether the partner is an openFT or FTP or ADM partner. For more information on address specifications see section “Defining partner properties” on page 108.

SECURITY-LEVEL =
Assigns a security level to a remote system.

SECURITY-LEVEL = *STD
If you set this operand to *STD or if you do not enter a value here, a standard security level is assigned to the remote system. This standard security level is defined using the command MODIFY-FT-OPTIONS. You can define a fixed value or specify that the value should be attribute-dependent.

SECURITY-LEVEL = *BY-PARTNER-ATTRIBUTES
If you set this operand to *STD or if you do not enter a value here, a standard security level is assigned to the remote system:

- This setting assigns partners that are authenticated by openFT the security level 10.
- Partners known to the transport system (e.g. VTAM or DNS) are assigned security level 90.
- All other partners are assigned security level 100.

SECURITY-LEVEL = <integer 1..100>
Must be specified if you wish to assign an individual security level to a specific remote system.

STATE =
You can use this operand to control locally distributed FT requests which are directed to the remote system you have just entered.

STATE = *ACTIVE(...)
Deactivated locally distributed FT requests to this remote system are processed.

AUTOMATIC-DEACT =
Defines whether cyclical attempts to establish a connection to a specific partner system are prohibited after a number of attempts by deactivating the partner system.

AUTOMATIC-DEACT = *NO
Failed attempts to establish a connection of this partner system do not result in its deactivation.

AUTOMATIC-DEACT = *YES
Failed attempts to establish a connection of this partner system result in its deactivation. If file transfer is to be resumed with this partner system, it must be explicitly reactivated.

STATE = *DEACT
Locally distributed FT requests to this remote system are not processed (not started) for the moment.

IDENTIFICATION =
Network-wide, unique identification of the openFT instance in the partner system.

IDENTIFICATION = *STD
For openFT and FTADM partners, the partner address or the hostname from the partner address is used as the identification. For FTP partners, no identification is set.

IDENTIFICATION = <composed-name 1..64> / <c-string 1..64 with-low>
The network-wide, unique instance ID of the openFT instance in the partner system. It is set by the FT administrator of the partner system (for example in BS2000, by using MODIFY-FT-OPTIONS IDENTIFICATION=, in Unix systems or Windows systems, by using ftmodo -id). The uniqueness of this ID must be based on something other than case-sensitivity. An instance ID may be comprised of alphanumeric characters or special characters. It is advisable only to use the special characters ".", "-", ":" or "%". The initial character must be alphanumeric or the special character "%". The "%" character may only be used as an initial character. An alphanumeric character must follow the "." character.
For more details on allocating instance IDs, please refer to section "Instance identification" on page 114.

SESSION-ROUTING-INFO =
If the partner system is only accessible by a go-between instance (for example openFTIF gateway), specify the address information that the gateway instance uses for re-routing here.

SESSION-ROUTING-INFO = *NONE
By default, no specification is required.
The session selector can be specified as a part of the partner address.

SESSION-ROUTING-INFO = *IDENTIFICATION
Connections to the partner are re-routed via a gateway that supports the instance ID as address information.

SESSION-ROUTING-INFO = <alphanum-name 1..8>
Connections to the partner are re-routed via a gateway that supports the specified character string as address information.

PARTNER-CHECK =
Modifies the global settings for the sender check in a partner-specific way. These settings are only valid for openFT partners that do not work with authentication (for example, partners with openFT V8.0 or earlier).

PARTNER-CHECK = *BY-FT-OPTIONS
The global settings are valid for the partners.

PARTNER-CHECK = *STD
Disables the expanded sender checking. The transport address of the partner is not checked, even if the expanded sender checking is globally enabled (see the FTMODOPT command).

PARTNER-CHECK = *TRANSPORT-ADDRESS
Enables the expanded sender checking. The transport address is checked, even if the expanded sender checking is globally disabled (see the FTMODOPT command).
If the transport address under which the partner is reporting does not correspond to the entry in the partner list, the request is rejected.

TRACE =
Allows you to modify the global settings for partner selection in the openFT trace function on a partner-specific basis.

TRACE = *BY-FT-OPTIONS
The global settings apply for the partner.

TRACE = *ON
The trace function is activated for this partner. However, the trace is only written if the global openFT trace function is also activated (see also the FTMODOPT command, TRACE option, SWITCH=*ON). The setting made here takes priority over the setting in the operating parameters for selecting partners for the monitoring function. See the option TRACE=(...,PARTNER-SELECTION=).

TRACE = *OFF
The trace function is deactivated for this partner.

AUTH-MANDATORY =
Allows you to force the authentication of a partner.

AUTH-MANDATORY = *NO
Authentication is not forced, i.e. this partner is not restricted with regard to authentication.

AUTH-MANDATORY = *YES
Authentication is forced, i.e. connections to and from this partner are only permitted with authentication.

PRIORITY=
This operand allows you to specify the priority of a partner in respect of processing requests that have the same request priority. This means that the partner priority only applies in the case of requests that have the same request priority, but that are issued to partners with a different partner priority.

PRIORITY = *NORMAL
The partner has normal priority.

PRIORITY = *LOW
The partner has low priority.

PRIORITY = *HIGH
The partner has high priority.

If the FTADDPTN command is executed correctly then no message is output.

6.9.1 Notes on entering partner systems

- You can enter the local system as a "remote" system in your own partner list. However, when performing file transfers with this system, you should note that files can be destroyed by being copied to themselves.
- It is advisable to store the FTADDPTN commands required for the entries in the partner list in a PS data set or in a PO/PDSE data set member. This facilitates the transition to a new partner list. You can generate such a file for an existing network description file using the parameter LAYOUT=*ZOS-PROC in the FTSHWPTN command.
- In large networks, especially in client-server configurations, it is a tiresome jog to enter individually in the partner list all the partner systems which are to communicate with the local system. In order to reduce this effort, *open*FT provides with the dynamic partners option for handling file transfer and file management jobs initiated in partner systems, but which have no separate entry in the partner list (see section section “Administering partners” on page 108).

6.9.2 Sample partner system entries

As of *open*FT V10 for z/OS, the file for the partner list is created when *open*FT is started and does not have to be created by issuing a command any longer.

The following examples demonstrate how various partner systems, accessed using a variety of transport systems, are successively entered into the partner list.

In these examples it is assumed that the local system possesses the FT identifier *ZOS1*. All partner systems that use Network Description Files or partner lists containing symbolic names for partner systems (i.e. remote *open*FT for z/OS systems and *open*FT for BS2000 systems) address the local *open*FT system under the symbolic name *FTZOS1*. Although this is not really necessary from a technical point of view (symbolic names do not need to be consistent throughout the network; they simply have to be unique within the partner list on each individual *open*FT instance), it helps to make the examples easier to follow.

The examples below have been harmonized with other examples presented in this manual, i.e.:

- The examples relating to SNA interconnection have been harmonized with the examples relating to the generation of the transport system in the section “openFT interconnection via an SNA network” on page 25.
- The examples relating to TCP/IP interconnection have been harmonized with the example for the TNSTCPIP member of the FT parameter library in the section “Setting up the FT parameter library” on page 49.

1. A partner system with *open*FT as of V8.1 and the symbolic name *XAS1* is to be directly connected to the local system via TCP/IP. The instance identifier is *VAR2.MOULINET.FR*.

   If the partner's Internet address has been assigned to a host name (in the example: *XAS123*) in the z/OS name services, the remote *open*FT system can be entered in the local system's partner list using the following command:

   ```
   FTADDPTN PARTNER-NAME=XAS1,PARTNER-ADDRESS=XAS123
   ,IDENTIFICATION='VAR2.MOULINET.FR'
   ```

   This example functions for z/OS, BS2000, Unix and Windows partner systems, if the main station of the remote *open*FT system has been assigned the transport selector *$FJAM* and the port number *1100* there. These are the recommended default values in all *open*FT systems. Divergent values can be specified for the transport selector and the port number using the parameter PARTNER-ADDRESS in the FTADDPTN command, for example for port number 1111 and T selector TSELOPFT:

   ```
   PARTNER-ADDRESS=XAS123:1111.TSELOPFT
   ```

2. The partner system with the symbolic name *FTZOS1*, a partner with *open*FT V9.0 for z/OS, is to be entered in the partner list. The connection to the partner is established via SNA. It has the ftid *ZOS1* and the instance identifier *VAR1.FUSINET.AT*. The corresponding command is:

   ```
   FTADDPTN PARTNER-NAME=FTZOS1,PARTNER-ADDRESS=FJMZOS1:SNA
   ,IDENTIFICATION='VAR1.FUSINET.AT')
   ```

   In the case of partner systems with *open*FT V8.0 (or earlier), the instance identifier is derived from the processor name specifications and the partner system's *open*FT main station (usually $FJAM).

3. A partner system with *open*FT as of V8.1 for Unix systems and the symbolic name *FTUNIX1* is to be connected via SNA and TRANSIT-SERVER / TRANSIT-CLIENT. The LU name of the Unix partner system is *FJML0717* and its instance identifier is *UX.FUSINET.AT*.

   ```
   FTADDPTN PARTNER-NAME=FTUNIX1,PARTNER-ADDRESS=FJML0717:SNA
   ,IDENTIFICATION='UX.FUSINET.AT'
   ```

4. An FTP partner system using *open*FT for Unix systems as of V10 and the symbolic name *FTPX* is to be connected over TCP/IP. The host name of the partner system is *FTPHOST1* and the default port number 21 is to be used.

   ```
   FTADDPTN PARTNER-NAME=FTPX,PARTNER-ADDRESS=FTP://FTPHOST1
   ```

5. The partner system SERVER11 with openFT V11 for Unix systems is a remote administration server. The default port number (11000) is to be used for remote administration. The partner address is to be used for identification:

   ```
   FTADDPTN PARTNER-NAME=ADMINSRV,PARTNER-ADDRESS=FTADM://SERVER11
   ```

6.10 FTADM Execute remote administration command

Note on usage

User group: Users configured as remote administrators on the remote administration server.

A remote administration server must be deployed in order to use this command.

The command can be specified under TSO.

Description of the function

The FTADM command allows you to act as a remote administrator and administer an openFT instance via a remote administration server. The remote administration server accepts the administration request, checks the authorization and forwards the request to the openFT instance that is to be administered.

In addition, as remote administrator, you can use FTADM command to query the following information from the remote administration server (see page 209):

- You can determine what openFT instances you are authorized to administer and what remote administration permissions you have for these instances.
- You can read the ADM traps that the openFT instances you are administering have sent to the remote administration server. For this to be possible, the remote administration server must also be configured as an ADM trap server for the administered openFT instances. For details, see the section “ADM traps” on page 162.

Format

```
FTADM

 PARTNER-SERVER = <text 1..200 with-low>
,TRANSFER-ADMISSION = <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
,ROUTING-INFO = <text 1..200 with-low> / <c-string 1..200 with-low> / *NONE
,CMD = <c-string 1..1800 with-low>
,OUTPUT =*STDERR / *STDOUT / *FILE(...)
    *FILE(...)
     |     FILE-NAME = <filename 1..59>
,DATA-ENCRYPTION = *NO / *YES
```

Operands

PARTNER-SERVER= <text 1..200 with-low>
Specifies the partner name in the partner list or the address of the remote administration server. The remote administration server must be addressed as an ADM partner. For details, see the section section “Defining partner properties” on page 108.

TRANSFER-ADMISSION =
Specifies the FTAC transfer admission for accessing the remote administration server.

ROUTING-INFO =
Contains the routing information required to forward the remote administration command from the remote administration server to the required openFT instance.

ROUTING-INFO = <text 1..200 with-low> / <c-string 1..200 with-low>
Specifies the pathname of the openFT instance that you want to administer. The pathname is configured on the remote administration server by the ADM administrator. You can get the pathname by running the command ftshwc on the remote administration server. See the section “Determining the names of the openFT instances” on page 160.

ROUTING-INFO = *NONE
No routing information is required, i.e. the command is executed directly on the remote administration server. Only specific commands, however, (ftshwc and ftshwatp) can be executed directly on the remote administration server. You will find a brief description of these commands on page 209.

CMD =
Remote administration server command in the syntax of the openFT instance to be administered. A remote administration command can only be processed if the remote system is using an FT product that supports this function (see the section “Remote administration commands” on page 205).

CMD = <c-string 1..1800 with-low>
The remote administration command to be executed.

OUTPUT =
Specifies where the data generated by the command should be output following transfer in the local system.

OUTPUT = *STDERR
The data is written to *STDERR.

OUTPUT = *STDOUT
The data is written to *STDOUT.

OUTPUT = *FILE(...)
The data is written to a file. Please note that a previously specified command has to use *STDOUT (z/OS) or *SYSLST (BS2000) for the output channel since the output will otherwise not be written to the file.

FILE-NAME = <filename 1..59>
Name of the output file.

DATA-ENCRYPTION =
Specifies whether the data is to be transferred in encrypted form. The encryption of the request description data is not affected by this parameter.

DATA-ENCRYPTION = *NO
The data is transferred unencrypted.

DATA-ENCRYPTION = *YES
The data is transferred encrypted.

6.10.1 Remote administration commands

The following tables list the possible remote administration commands on the individual openFT platforms and on the remote administration server. The Permission column shows the permission required to execute the command as a remote administration command. The following permissions are possible:

FTOP	Read FT access (FT operator)
FT	Read and modify FT access (FT administrator)
FTAC	Read and modify FTAC access (FTAC administrator)

If a number of permissions are specified, e.g. FT | FTAC, it is sufficient if one of these permissions applies, i.e. FT or FTAC.

In the case of a remote administration request, these permissions are compared with the permissions you have on the relevant instance as a remote administrator. The ADM administrator defines the permissions in the configuration data of the remote administration server. If your permissions are not sufficient, the request is rejected and an appropriate message is issued.

Commands for openFT partners in BS2000

The commands have to be prefixed with "\" (backslash) before the command name.

BS2000 command	Short forms and aliases	Permission
ADD-FT-PARTNER	ADD-FT-PART FTADDPTN	FT
CANCEL-FILE-TRANSFER	CAN-FILE-T, CNFT NCANCEL, NCAN FTCANREQ	FT
CREATE-FT-KEY-SET	CRE-FT-KEY FTCREKEY	FT
CREATE-FT-PROFILE	CRE-FT-PROF	FTAC
DELETE-FT-KEY-SET	DEL-FT-KEY FTDELKEY	FT
DELETE-FT-LOGGING-RECORDS	DEL-FT-LOG-REC FTDELLOG	FT	FTAC
DELETE-FT-PROFILE	DEL-FT-PROF	FTAC
MODIFY-FILE-TRANSFER	MOD-FILE-T FTMODREQ	FT
MODIFY-FT-ADMISSION-SET	MOD-FT-ADM	FTAC

BS2000 command	**Short forms and aliases**	**Permission**
MODIFY-FT-OPTIONS	MOD-FT-OPT FTMODOPT	FT
MODIFY-FT-PARTNER	MOD-FT-PART FTMODPTN	FT
MODIFY-FT-PROFILE	MOD-FT-PROF	FTAC
REMOVE-FT-PARTNER	REM-FT-PART FTREMPTN	FT
SHOW-FILE-TRANSFER	SHOW-FILE-T, SHFT NSTATUS, NSTAT FTSHWREQ	FT | FTOP
SHOW-FT-ADMISSION-SET	SHOW-FT-ADM-S	FTAC
SHOW-FT-DIAGNOSTIC	SHOW-FT-DIAG FTSHWD	FT | FTOP | FTAC
SHOW-FT-INSTANCE	SHOW-FT-INST	FT | FTOP
SHOW-FT-LOGGING-RECORDS	SHOW-FT-LOG-REC FTSHWLOG	FT | FTOP | FTAC
SHOW-FT-MONITOR-VALUES [1]	SHOW-FT-MON-VAL FTSHWMON	FT | FTOP
SHOW-FT-OPTIONS	SHOW-FT-OPT FTSHWOPT	FT | FTOP
SHOW-FT-PARTNERS	SHOW-FT-PART FTSHWPTN	FT | FTOP
SHOW-FT-PROFILE	SHOW-FT-PROF	FTAC
START-FTTRACE	FTTRACE	FT | FTOP
STOP-FT	FTSTOP	FT
UPDATE-FT-PUBLIC-KEYS	UPD-FT-PUB-KEY FTUPDKEY	FT

[1] As of V11.0

Commands for openFT partners in z/OS

z/OS command	Alias	Permission
FTADDPTN		FT
FTCANREQ	NCANCEL, NCAN	FT
FTCREKEY		FT
FTCREPRF		FTAC
FTDELKEY		FT
FTDELLOG		FT | FTAC
FTDELPRF		FTAC
FTHELP		FT | FTOP | FTAC
FTINFO		FT | FTOP | FTAC
FTMODADS		FTAC
FTMODOPT		FT
FTMODPRF		FTAC
FTMODPTN		FT
FTMODREQ		FT
FTREMPTN		FT
FTSHWADS		FTAC
FTSHWD		FT | FTOP | FTAC
FTSHWINS		FT | FTOP
FTSHWLOG		FT | FTOP | FTAC
FTSHWMON [1]		FT | FTOP
FTSHWNET		FT | FTOP
FTSHWOPT		FT | FTOP
FTSHWPRF		FTAC
FTSHWPTN		FT | FTOP
FTSHWREQ	NSTATUS, NSTAT	FT | FTOP
FTSTOP		FT
FTTRACE		FT | FTOP
FTUPDKEY		FT

[1] As of V11.0

Commands for openFT partners in Unix and Windows systems

Command	Comment	Permission
fta	up to V10.0	FT
ftaddptn		FT
ftc	up to V10.0	FT
ftcanr		FT
ftcans	openFT-Script command	FT
ftcrek		FT
ftcrep		FTAC
ftdelk		FT
ftdell		FT I FTAC
ftdelp		FTAC
ftdels	openFT-Script command	FT
fthelp		FT I FTOP I FTAC
fti	up to V10.0	FT I FTOP
ftinfo		FT I FTOP I FTAC
ftmoda		FTAC
ftmodo		FT
ftmodp		FTAC
ftmodptn		FT
ftmodr		FT
ftping		FT I FTOP
ftremptn		FT
ftrs	up to V10.0	FT
ftsetpwd	Windows systems only	FT I FTOP
ftshwa		FTAC
ftshwact	openFT-Script command	FT I FTOP
ftshwd		FT I FTOP I FTAC
ftshwi		FT I FTOP
ftshwl		FT I FTOP I FTAC
ftshwm	as of V11.0	FT I FTOP
ftshwo		FT I FTOP
ftshwp		FTAC
ftshwptn		FT I FTOP

Command	**Comment**	**Permission**
ftshwr		FT I FTOP
ftshws	openFT-Script command	FT I FTOP
ftstop		FT
fttrace		FT I FTOP
ftupdk		FT

Commands on the remote administration server

FTADM allows you to execute the commands *ftshwc* and *ftshwatp* on the remote administration server. To do this, you must specify ROUTING-INFO=*NONE:

Command	**Comment**	**Permission**
ftshwc	Gets the instances that the remote administrator is permitted to administer.	FT I FTOP I FTAC (I.e. all instances are displayed for which the remote administrator has this permission.)
ftshwatp	Outputs the ADM traps of the openFT instances that can be administered.	FT I FTOP (I.e. ADM traps of all instances are displayed for which the remote administrator has this permission.)

These commands also provide further options. For details, see, for instance, the manual "openFT V11.0 for Unix Systems - Installation and Administration".

6.11 FTCREKEY
Create a key pair set

Note on usage

User group: FT administrator

You can issue the FTCREKEY command under TSO with the FT system running.

Functional description

Using this FTCREKEY command, you create a key pair for authenticating your openFT instance in partner systems (RSA procedures). The key pair consists of a private key, administered internally by openFT, and a public key.

Public keys are stored under the name:

<openft qualifier>.<inst>.SYSPKF.R<key reference>.L<key length>

Here, the first two name parts are replaced by OPENFT QUALIFIER and the instance name.

The key reference is a numerical designator for the version of the key pair. The key length is 768 or 1024 or 2048. The three key lengths are always generated. The public key files are text files which are created in the character code of the respective operating system, i.e. EBCDIC.DF04-1 for BS2000, IBM1047 for z/OS, ISO8859-1 for Unix systems and CP1252 for Windows systems.

In a file <openft qualifier>.<inst>.SYSPKF.COMMENT you can store comments, which are written in the first lines of the public key files when a key pair set is created. Such comments could be, for example, the communications partner and the telephone number of the FT administrator on duty. The lines in the SYSPKF.COMMENT file may be a maximum of 78 characters long.

So that your openFT instance can be authenticated by partner systems (using openFT as of version 8.1), the public key file must be transported to the partners via a reliable path and re-coded if necessary (see section “Authentication” on page 113).

In order to make an authorized update of the key pair sets, openFT supports up to three key pair sets at a time.

The most current key pair is used for delivering the session key for encrypting user data and request description data. If there is no key pair set, work proceeds without encryption.

Format

FTCREKEY

Without operands

In the event of an error (three key pair sets already exist), the following message is output:

```
FTR1029 OPENFT: Maximum number of key pairs exceeded
```

6.12 FTCREPRF Create admission profile

Note on usage

User group: FTAC user and FTAC administrator

A prerequisite for using this command is the use of openFT-AC.

Functional description

All FTAC users can use FTCREPRF to set up their own admission profiles under their user IDs. Users must activate admission profiles predefined by the FTAC administrator with FTMODPRF (see page 273ff) before they can be used. Profiles predefined by the FTAC administrator may be used immediately if the FTAC administrator also possesses the "SU" privilege.

The FTAC administrator can use FTCREPRF to create admission profiles for each user. It is necessary to distinguish between three cases:

- The FTAC administrator possesses the"SU" privilege (see page 62). He can then create profiles for other user IDs without restriction and these profiles are available for immediate use.
- If the FTAC administrator does not possess the "SU" privilege but specifies ACCOUNT and PASSWORD in the USER-ADMISSION parameter, then he may also assign a TRANSFER-ADMISSION for the profile. However, this functions only for as long as the current password for the user ID corresponds to the one defined in the profile.
- If the FTAC administrator does not possess the "SU" privilege and also does not specify the user’s password then he may not define any TRANSFER-ADMISSION in the profile. In this case, the user must then assign the profile a TRANSFER-ADMISSION with the FTMODPRF command.

It is possible to create an admission profile for "pre-processing" or "post-processing". To do this, the FILE-NAME operand must start with the pipe symbol '|'. After this has been done, one or more TSO commands can be specified. For detailed information refer to the section "Preprocessing and postprocessing" in the User Guide.

Format

(part 1 of 2)

```
FTCREPRF

 NAME = *STD / <alphanum-name 1..8>
,PASSWORD = *NONE / <alphanum-name 1..8>
,TRANSFER-ADMISSION = *NOT-SPECIFIED / <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) /
                      <x-string 15..64>(...)

   <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
     |   VALID = *YES / *NO
     |  ,USAGE = *PRIVATE / *PUBLIC
     |  ,EXPIRATION-DATE = *NOT-RESTRICTED / <date 8..10>
,PRIVILEGED = *NO / *YES
,IGNORE-MAX-LEVELS = *NO / *YES / *PARAMETERS(...)

   *PARAMETERS(...)
     |   OUTBOUND-SEND = *NO / *YES
     |  ,OUTBOUND-RECEIVE = *NO / *YES
     |  ,INBOUND-SEND = *NO / *YES
     |  ,INBOUND-RECEIVE = *NO / *YES
     |  ,INBOUND-PROCESSING = *NO / *YES
     |  ,INBOUND-MANAGEMENT = *NO / *YES
,USER-ADMISSION = *OWN / *PARAMETERS(...)

   *PARAMETERS(...)
     |   USER-IDENTIFICATION = *OWN / <name 1..8>
     |  ,ACCOUNT = *OWN / *NOT-SPECIFIED / <alphanum-name 1..40> / <c-string 1..40>
     |  ,PASSWORD = *OWN / *NOT-SPECIFIED / <alphanum-name 1..8> / *NONE
,INITIATOR = (*LOCAL, *REMOTE) / list-poss(2): *LOCAL / *REMOTE /
,TRANSFER-DIRECTION = *NOT-RESTRICTED / *FROM-PARTNER / *TO-PARTNER
,PARTNER = *NOT-RESTRICTED / list-poss(50): <text 1..200 with-low>
,MAX-PARTNER-LEVEL = *NOT-RESTRICTED / <integer 0..100>
,FILE-NAME = *NOT-RESTRICTED / <filename1..59> / <c-string 1..512 with-low> / *EXPANSION(...)

  ,*EXPANSION(...)
     |   PREFIX = <filename 1..58> / <filename-prefix 2..50> / <c-string 1..511 with-low>
```

(part 2 of 2)

```
,FILE-PASSWORD = *NOT-RESTRICTED / *NONE / <alphanum-name 1..8>
,PROCESSING-ADMISSION = *SAME / *NOT-RESTRICTED / *PARAMETERS(...)
   *PARAMETERS(...)
     |   USER-IDENTIFICATION = *SAME / *NOT-RESTRICTED / <name 1..8>
     |  ,ACCOUNT = *SAME / *NOT-RESTRICTED / <alphanum-name 1..40> / <c-string 1..40>
     |  ,PASSWORD = *SAME / *NOT-RESTRICTED / *NONE / <alphanum-name 1..8>
,SUCCESS-PROCESSING = *NOT-RESTRICTED / *NONE / <c-string 1..1000 with-low> / *EXPANSION(...)
   *EXPANSION(...)
     |   PREFIX = *NOT-RESTRICTED / <c-string 1..999 with-low>
     |  ,SUFFIX = *NOT-RESTRICTED / <c-string 1..999 with-low>
,FAILURE-PROCESSING = *NOT-RESTRICTED / *NONE / <c-string 1..1000 with-low> / *EXPANSION(...)
   *EXPANSION(...)
     |   PREFIX = *NOT-RESTRICTED / <c-string  1..999 with-low>
     |  ,SUFFIX = *NOT-RESTRICTED / <c-string 1..999 with-low>
,WRITE-MODE =  *NOT-RESTRICTED / *NEW-FILE / *REPLACE-FILE / *EXTEND-FILE
,FT-FUNCTION = *NOT-RESTRICTED / list-poss(5): *TRANSFER-FILE / *MODIFY-FILE-ATTRIBUTES /
                     *READ-DIRECTORY / *FILE-PROCESSING / *REMOTE-ADMINISTRATION
,USER-INFORMATION = *NONE / <c-string 1..100 with-low>
,DATA-ENCRYPTION = *NOT-RESTRICTED / *NO / *YES
```

Operands

NAME = <alphanum-name 1..8>

With NAME, the admission profile is given a name. This name must be unique among all admission profiles on that user ID. If an admission profile with this name already exists, FTAC rejects the command with the message:

```
FTC0100 COMMAND REJECTED. FT-PROFILE ALREADY EXISTS
```

The command FTSHWPRF (see page 346ff) can be used to view the already existing names. To obtain this information, the command FTSHWPRF can be entered without operands.

NAME = *STD

Creates a default admission profile for the user ID. You must specify *NOT-SPECIFIED as the transfer admission, because a default admission profile in a request is addressed using the user ID and password. You must not specify the parameters VALID, USAGE and EXPIRATION-DATE for a default admission profile.

PASSWORD =
FTAC password which authorizes you to issue FTAC commands on your user ID, if such a password was defined in your admission set.

PASSWORD = <u>*NONE</u>
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
This FTAC password is required.

TRANSFER-ADMISSION =
With TRANSFER-ADMISSION, you define transfer admission. If this transfer admission is entered in an FT request instead of the LOGON admission, then the access rights are valid which are defined in this admission profile. This transfer admission must be unique in the entire openFT instance, so that there is no conflict with other transfer admissions which other FTAC users have defined for other access rights. When the transfer admission which you have selected has already been used, then FTAC rejects the command with the message:

```
FTC0101 COMMAND REJECTED. TRANSFER-ADMISSION ALREADY EXISTS
```

The FTAC administrator can also assign a transfer admission when he creates an admission profile for a user ID. To do this, the complete USER-ADMISSION must be entered for the user ID in question (USER-IDENTIFICTATION, ACCOUNT and PASSWORD) or the FTAC administrator must have the SU privilege.

TRANSFER-ADMISSION = *NOT-SPECIFIED
This entry is used to set up a profile without transfer admission. If the profile is not a default admission profile, it is locked until you specify a valid transfer admission or the owner specifies a valid transfer admission.

TRANSFER-ADMISSION = <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
The character string must be entered as the transfer admission in the transfer request. The alphanumeric entry is always stored in lower-case letters.

VALID = <u>*YES</u>
The transfer admission is valid.

VALID = *NO
The transfer admission is not valid. With this entry, users can be denied access to the profile.

USAGE = <u>*PRIVATE</u>
Access to your profile is denied for security reasons, when someone with another user ID attempts a second time to specify the TRANSFER ADMISSION which has already been used by you.

USAGE = *PUBLIC
Access to your profile is not denied if another user happens to “discover” your TRANSFER-ADMISSION. “Discovery” means that another user ID attempted to specify the same TRANSFER ADMISSION twice. This is rejected for security reasons.

EXPIRATION-DATE = *NOT-RESTRICTED
The use of this transfer admission is not restricted with respect to time.

EXPIRATION-DATE = <date 8..10>
The use of the transfer admission is only possible until the given date (max. 19.01.2038). The entry must be made in the form yyyy-mm-dd or yy-mm-dd.

PRIVILEGED =
The FTAC administrator can privilege the profile. FT requests which are processed with a privileged admission profile are not subject to the restrictions which are set for MAX-ADM-LEVEL (see page 306) in the admission set.

PRIVILEGED = *NO
The admission profile is not privileged.

PRIVILEGED = *YES
The admission profile is privileged.
Only the FTAC administrator can use this entry.

IGNORE-MAX-LEVELS =
You can determine for which of the six basic functions the restrictions of the admission set should be ignored. The user’s MAX-USER-LEVELS can be exceeded in this way. The MAX-ADM-LEVELS in the admission set can only be effectively exceeded with an admission profile which has been designated as privileged by the FTAC administrator. The FTAC user can set up an admission profile for himself for special tasks (e.g. sending a certain file to a partner system with which he normally is not allowed to conduct a file transfer), which allows him to exceed the admission set. This profile must be explicitly given privileged status by the FTAC administrator.
If you enter IGNORE-MAX-LEVELS=*YES, the settings for **all** the basic functions are ignored. If you wish to ignore the admission set for **specific** basic functions, you need to do this with the operands explained later in the text.

The following table shows which partial components of the file management can be used under which conditions:

Inbound file management function	**Setting in admission set/extension in profile**
Show file attributes	Inbound sending (IBS) permitted
Modify file attributes	Inbound receiving (IBR) **and** Inbound file management (IBF) permitted
Rename files	Inbound receiving (IBR) **and** Inbound file management (IBF) permitted
Delete files	Inbound receiving (IBR) permitted **and** write rule = overwrite in profile
Show directories	Inbound file management (IBF) permitted **and** direction = to partner in profile
Create, rename, delete directories	Inbound file management (IBF) permitted **and** direction = from partner in profile

IGNORE-MAX-LEVELS = <u>*NO</u>
FT requests which are processed with the admission profile are subject to the restrictions of the admission set.

IGNORE-MAX-LEVELS = *YES
*YES allows you to communicate with partner systems whose security level exceeds the specifications of the admission set. Unless you have a privileged profile, you can only exceed the MAX-USER-LEVELS and not the MAX-ADM-LEVELS in the admission set. You must respect the restrictions defined in the admission set by the FTAC administrator. The SHOW-FT-ADMISSION-SET command provides information on the entries made by the FTAC administrator (see example on page 306. This includes information about the current MAX-USER-LEVELS and MAX-ADM-LEVELS settings.

IGNORE-MAX-LEVELS = *PARAMETERS(...)
The following operands can be used to selectively deactivate the default settings for the individual basic functions.

> **OUTBOUND-SEND = <u>*NO</u>**
> The maximum security level which can be reached with the basic function “outbound send” is determined by the admission set.
>
> **OUTBOUND-SEND = *YES**
> For the basic function “outbound send”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.
>
> **OUTBOUND-RECEIVE = <u>*NO</u>**
> The maximum security level which can be reached with the basic function “outbound receive” is determined by the admission set.

OUTBOUND-RECEIVE = *YES
For the basic function “outbound receive”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-SEND = <u>*NO</u>
The maximum security level which can be reached with the basic function “inbound send” is determined by the admission set.

INBOUND-SEND = *YES
For the basic function “inbound send”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The same applies to the partial component “display file attributes” of the basic function “inbound file management”.

INBOUND-RECEIVE = <u>*NO</u>
The maximum security level which can be reached with the basic function “inbound receive” is determined by the admission set.

INBOUND-RECEIVE = *YES
You can disregard your settings for “inbound receive” in the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The same applies to the partial components of the basic function “inbound file management”:

- delete files, as long as the file attributes are set accordingly,
- modify file attributes, if the basic function “inbound file management” was admitted in the admission set or in the admission profile.

INBOUND-PROCESSING = <u>*NO</u>
The maximum security level which can be reached with the basic function “inbound follow-up processing” is determined by the admission set.

INBOUND-PROCESSING = *YES
For the basic function “inbound follow-up processing”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-MANAGEMENT = <u>*NO</u>
The maximum security level which can be reached with the basic function “inbound file management” is determined by the admission set.

INBOUND-MANAGEMENT = *YES
For the basic function “inbound file management”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The partial component “modify file attributes” of the basic function “inbound file management” only functions if the basic function “inbound receive” was admitted in the admission set or admission profile.

USER-ADMISSION =
User ID under which the profile is to be saved. FT requests which work with this admission profile access the given user ID in the local system.
If, as FTAC administrator, you create the admission profile for a user, you cannot generally specify either ACCOUNT or PASSWORD in the USER-ADMISSION operand (since these should be known only to the user in question). These specifications must be entered by the user by means of MODIFY-FT-PROFILE before the profile can actually be used. If, as FTAC administrator, you also possess the SU privilege, then you can also create a profile which is available for immediate use even without a password specification.
If you want to assign a transfer admission to a user's admission profile then you must specify both the USER-ADMISSION and the ACCOUNT and PASSWORD.

USER-ADMISSION = *OWN
For USER-IDENTIFICATION and ACCOUNT, the specifications for your user ID and your account number are taken from your LOGON authorization. A possible z/OS password is only taken from your LOGON authorization when an FT request accesses the admission profile.

Admission profiles in which USERID, ACCOUNT and PASSWORD are set to their default values via *OWN cannot be used for pre-processing, post-processing or follow-up processing. For pre-processing and post-processing, these parameters must be explicitly assigned a value in USER-ADMISSION. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

USER-ADMISSION = *PARAMETERS(...)
You can also enter the individual components of the user ID. This allows you to keep FT requests which use this admission profile under a different account number, for example. Or, a password can be set in the admission profile. FT requests which use this admission profile will then only function if their current LOGON password corresponds to the pre-set password.

USER-IDENTIFICATION =
Your user ID in z/OS.

USER-IDENTIFICATION = *OWN
Your user ID is taken from the LOGON authorization.

USER-IDENTIFICATION = <name 1..8>
User ID associated with the profile. The FTAC administrator can also specify foreign user IDs.

ACCOUNT =
Account number under which an FT request is to be kept when it uses this admission profile.

ACCOUNT = *OWN
The account number is taken from your LOGON authorization.

ACCOUNT = *NOT-SPECIFIED
No account number is defined.

ACCOUNT = <alphanum-name 1..8>
An FT request should be kept under the account number specified when it accesses this admission profile. You can enter any account number which is associated with the user ID.
You can also specify accounting information which contains the account number to be used.

PASSWORD =
z/OS password associated with your user ID.

PASSWORD = *OWN
When an FT request refers to this admission profile, FTAC uses the BS2000 password valid for your user ID at that moment. This prevents you from having to modify the admission profile if the BS2000 password is changed.

PASSWORD = *NOT-SPECIFIED
The password will be entered by the owner of the admission profile. This function allows the FTAC administrator to create profiles for foreign user IDs.

PASSWORD = *NONE
No BS2000 password is required for the user ID.

PASSWORD = <alphanum-name 1..8>
When an FT request accesses the admission profile, the password specified is compared with the current LOGON password. If the two do not correspond, the FT request is rejected.

INITIATOR =
Determines if initiators from local and/or remote systems are permitted to use this admission profile for their FT requests.

INITIATOR = (*LOCAL,*REMOTE)
This admission profile may be used by initiators from local and remote systems.

INITIATOR = *REMOTE
This admission profile may only be used for FT requests by initiators from remote systems.

INITIATOR = *LOCAL
This admission profile may only be used for FT requests by initiators from the local system.

TRANSFER-DIRECTION =
Determines which transfer direction may be used with this admission profile. The transfer direction is always determined from the system in which the admission profile was defined.

TRANSFER-DIRECTION = *NOT-RESTRICTED
With this admission profile, files can be transferred to and from a partner system.

TRANSFER-DIRECTION = *FROM-PARTNER
With this admission profile, files can only be transferred from a partner system to your system. It is not possible to display file attributes/directories (partial components of “inbound file management”).

TRANSFER-DIRECTION = *TO-PARTNER
With this admission profile, files can only be transferred from your system to a partner system. It is not possible to modify file attributes or delete files (partial components of “inbound file management”).

PARTNER =
Specifies that this admission profile is to be used only for FT requests which are processed by a a certain partner system.

PARTNER = *NOT-RESTRICTED
The range of use for this admission profile is not restricted to FT requests with certain partner systems.

PARTNER = list-poss(50): <text 1..200 with-low>
The admission profile only permits those FT requests which are processed with the specified partner systems. A maximum of 50 partner names can be specified. The total length of all the partners may not exceed 1000 characters. You may specify the name from the partner list or the address of the partner system, see also section “Specifying partner addresses” on page 108. It is recommended, to use the name from the partner list. The format shown in the long form of the logging output provides an indication of how a partner address should be entered in an FTAC profile.

MAX-PARTNER-LEVEL =
A maximum security level can be specified. The admission profile will then only permit those FT requests which are processed with partner systems which have this security level or lower.
MAX-PARTNER-LEVEL works in conjunction with the admission set. When non-privileged admission profiles are used, the access check is executed on the basis of the smallest specified value.

MAX-PARTNER-LEVEL = *NOT-RESTRICTED
If FT requests are processed with this admission profile, then the highest accessible security level is determined by the admission set.

MAX-PARTNER-LEVEL = <integer 0..100>
All partner systems which have this security level or lower can be communicated with.

i When you set MAX-PARTNER-LEVEL=0, you prevent access to the admission profile (for the moment). No FT requests can be processed with this admission profile.

FILE-NAME =

Determines which files or library members under your user ID may be accessed by FT requests that use this admission profile.

FILE-NAME = *NOT-RESTRICTED

Permits unrestricted access to all files and library members of the user ID.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>

Only the specified file may be accessed.
However, openFT is also able to generate unique filenames automatically, thus providing an easy way of avoiding conflicts. This is done by specifying the string %UNIQUE at the end of the filename which is predefined here (see section “File names” in the User Guide). When follow-up processing is specified, this file can be referenced with %FILENAME, %FILN or %FILX (see User Guide).

You can also directly specify file transfer with file pre- or post-processing here by entering a pipe symbol ’|’ followed by TSO commands.

FILE-NAME = *EXPANSION(PREFIX = <filename 1..58> / <partial-filename 2..50> / <c-string 1..511 with-low>)

Restricts access to a number of files which all begin with the same prefix. If a filename is entered in an FT request which works with this admission profile, FTAC sets the *prefix* defined with EXPANSION in front of this filename. The FT request is then permitted to access the file *PrefixFilename*.

Example

– PREFIX=DAGOBERT.; an FT request in which FILE-NAME=BOERSE is specified, then accesses the file DAGOBERT.BOERSE.
– PREFIX=TOOLS.CLIST/; an FT request in which FILE-NAME=MEMBER01 is specified, then accesses the file TOOLS.CLIST(MEMBER01).

Please note that the part of a filename which is specified in the file transfer command still has to be of the type <filename>.

If you want to perform file transfer with pre- or post-processing, you should indicate this by entering the pipe symbol ‘|’ at the start of the prefix. The created FTAC profile can then be used only for file transfer with pre- or post-processing since the file name that is generated also starts with a ‘|’. The variable %TEMPFILE can also be used in the filename prefix. You can find detailed information on preprocessing and postprocessing in the section of the same name in the User Guide.

The maximum length of the entire pre- or post-processing command is limited to the maximum length of the file name. If several commands are specified, then they must be separated by a semicolon (‘;’).

Example

```
FILE-NAME = *EXP(C‘|Command1;Command2;Command3; ...‘)
```

If you specify a name prefix that starts with a pipe character with *EXP(PREFIX=...), the preprocessing or postprocessing command of the FT request must not contain any semicolons. If the preprocessing or postprocessing command nevertheless contains semicolons, it must be enclosed in '...' (single quotes) .

Special cases

– A file name or file name prefix that begins with the string '|ftexecsv' must be specified for admission profiles that are to be exclusively used for the ftexec command (see "Example 3" on page 230).

– Specify the file name prefix '|*ftmonitor' for admission profiles that are exclusively used for monitoring. A profile of this sort can then be used in the openFT Monitor or in an ft or ncopy command from a Windows or Unix system (see "Example 2" on page 230).

FILE-PASSWORD =
You can enter a password for files into the admission profile. The FTAC functionality then only permits access to files which are protected with this password and to unprotected files. When a FILE-PASSWORD is specified in an admission profile, the password may no longer be specified in an FT request which uses this admission profile. This allows you to permit access to certain files to users in remote systems, without having to give away the file passwords.

FILE-PASSWORD = *NOT-RESTRICTED
Permits access to all files. If a password is set for a file, then it must be specified in the transfer request.

FILE-PASSWORD = *NONE
Only permits access to files without file passwords.

FILE-PASSWORD = <alphanum-name 1..8>
Only permits access to files which are protected with the password specified and to unprotected files. The password which has already been specified in the profile may not be repeated in the transfer request. PASSWORD=*NONE would be entered in this case!

PROCESSING-ADMISSION =
You can enter a user ID in your z/OS system. Any follow-up processing of an FT request will be executed under this user ID. With PROCESSING-ADMISSION in the admission profile, you do not need to disclose your LOGON authorization to partner systems for follow-up processing.

i Admission profiles in which ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for follow-up processing. For follow-up processing, these parameters must be explicitly assigned a value either in USER-ADMISSION or in PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *SAME
For the PROCESSING-ADMISSION, the values of the USER-ADMISSION are used. If *SAME is entered here, then any FT request which uses this profile must also contain PROCESSING-ADMISSION=*SAME or PROCESSING-ADMISSION=*NOT-SPECIFIED.

PROCESSING-ADMISSION = *NOT-RESTRICTED
FT requests which use this admission profile may contain any PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *PARAMETERS(...)
You can also enter the individual components of the user ID. This allows you to keep FT requests which use this admission profile under a different account number, for example. Or, a password can be set in the admission profile. FT requests which use this admission profile will then only function if their current LOGON password corresponds to the pre-set password.

USER-IDENTIFICATION =
Identifies the user ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = *SAME
The USER-IDENTIFICATION is taken from the USER-ADMISSION.

USER-IDENTIFICATION = *NOT-RESTRICTED
The admission profile does not restrict the user ID for the follow-up processing.

USER-IDENTIFICATION = <name 1..8>
FT requests which are processed with this admission profile are only permitted follow-up processing under this user ID. If another user ID is entered here, the parameter PASSWORD must also be entered. PASSWORD=*SAME is then not valid.

ACCOUNT =
Account number for the follow-up processing.

ACCOUNT = *SAME
The account number is taken from the USER-ADMISSION.

ACCOUNT = *NOT-RESTRICTED
Account number in FT requests which work with the admission profile. The admission profile does not restrict the account with regard to follow-up processing.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Follow-up processing is to be settled under this account number.
You an also specify accounting information containing the account number to be used.

PASSWORD =
You specify, where applicable, the z/OS password for the user ID under which the follow-up processing is to be executed. Here, you can enter a PASSWORD when the user ID in question doesn’t have such a password (yet).

PASSWORD = *SAME
The value *SAME is only valid if the PROCESSING-ADMISSION refers to your own user ID. If PASSWORD=*OWN is entered on USER-ADMISSION, then the password valid at the time of the request is used for the PROCESSING-ADMISSION.

PASSWORD = *NOT-RESTRICTED
Specifies the password in FT requests which work with the admission profile. The admission profile does not restrict the password with regard to follow-up processing.

PASSWORD = *NONE
FT requests which use this admission profile can only initiate follow-up processing on user IDs without a password.

PASSWORD = <alphanum-name 1..8>
FT requests which use this admission profile may only initiate follow-up processing on user IDs which are protected with this password.

SUCCESS-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a successful data transfer.

SUCCESS-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand SUCCESS-PROCESSING may be used without restriction.

SUCCESS-PROCESSING = *NONE
The admission profile does not permit follow-up processing after successful data transfer.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
Commands which are executed in the local system after successful data transfer.
The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

SUCCESS-PROCESSING = *EXPANSION(...)
If a SUCCESS-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *NOT-RESTRICTED
The follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified suffix is added to a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

Note that blanks at the end of the specification are removed in the FT request, when the follow-up command is assembled. Therefore blanks that are needed here, must be included at the beginning of the specification for SUFFIX.

Example

If PREFIX='SEND ' and SUFFIX=',USER(USER1)' is specified and SUCC='''FILE TRANSFER OK''' is defined in the FT request, FT executes the command "SEND 'FILE TRANSFER OK',USER(USER1)" for follow-up processing.

FAILURE-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a failed data transfer.

FAILURE-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand FAILURE-PROCESSING may be used without restriction.

FAILURE-PROCESSING = *NONE
The admission profile does not permit follow-up processing after failed data transfer.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
Commands which are executed in the local system after failed data transfer.
The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

FAILURE-PROCESSING = *EXPANSION(...)
If a FAILURE-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = <u>*NOT-RESTRICTED</u>
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = <u>*NOT-RESTRICTED</u>
The follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified suffix is added to a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

WRITE-MODE =
Determines the WRITE-MODE specification which is valid for this FT request. WRITE-MODE is only effective if the receive file is in the same system as the admission profile definition.

WRITE-MODE = <u>*NOT-RESTRICTED</u>
In an FT request which accesses this admission profile, the operand WRITE-MODE may be used without restrictions.

WRITE-MODE = *NEW-FILE
In the FT request, *NEW-FILE, *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. If the receive file already exists, the transfer will be rejected.

WRITE-MODE = *REPLACE-FILE
In the FT request of openFT partners, only *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. With ftp partners, *NEW-FILE may also be entered if the file does not yet exist.

WRITE-MODE = *EXTEND-FILE
In the FT request, only *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE.

FT-FUNCTION =
Permits the restriction of the profile validity to certain FT functions (=file transfer and file management functions).

FT-FUNCTION = <u>*NOT-RESTRICTED</u>
The full scope of FT functions is available. For reasons of compatibility, the specification NOT-RESTRICTED means that FILE-PROCESSING REMOTE-ADMINISTRATION are not permitted! All other functions are permitted if this value is specified.

FT-FUNCTION = (*TRANSFER-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY,*FILE-PROCESSING, *REMOTE-ADMINISTRATION)
The following file transfer functions are available:

***TRANSFER-FILE**
The admission profile may be used for the file transfer functions “transfer files”, “view file attributes” and “delete files”.

***MODIFY-FILE-ATTRIBUTES**
The admission profile may be used for the file transfer functions “view file attributes” and “modify file attributes”.

***READ-DIRECTORY**
The admission profile may be used for the file transfer functions “view directories” and “view file attributes”.

***FILE-PROCESSING**
The admission profile may be used for the “pre-processing” and “post-processing” file transfer function. The “transfer files” function must also be permitted.

The *FILE-PROCESSING specification is of relevance only for FTAC profiles without a filename prefix. Otherwise the first character of the filename prefix determines whether only normal data transfer (no pipe symbol |) or only pre-processing and post-processing (pipe symbol |) are to be possible with this FTAC profile.

***REMOTE-ADMINISTRATION**
The admission profile is allowed to be used for the "remote administration" function. This allows a remote administrator to administer the openFT instance using this profile. *REMOTE-ADMINISTRATION may only be specified by the FT administrator or FTAC administrator.

USER-INFORMATION =
Here, you enter a text in the admission profile. This text is displayed with the command FTSHWPRF.

USER-INFORMATION = <u>*NONE</u>
No text is stored in the profile.

USER-INFORMATION = <c-string 1..100 with-low>
Here, you enter a character string containing user information.

DATA-ENCRYPTION =
Restricts the encryption option for user data.

DATA-ENCRYPTION = <u>*NOT-RESTRICTED</u>
The encryption option for user data is not restricted. Both encrypted and unencrypted file transfers are accepted.

DATA-ENCRYPTION = *NO
Only those file transfers which do not have encrypted user data are accepted, i.e. encrypted requests are rejected.
If the request is made in a BS2000 or z/OS, for example, it must be specified there in the NCOPY request DATA-ENCRYPTION=*NO.

DATA-ENCRYPTION = *YES
Only those file transfer requests that have encrypted user data are accepted, i.e. unencrypted requests are rejected.
If the request is made in a BS2000 or z/OS, for example, it must be specified there in the NCOPY request DATA-ENCRYPTION=*YES.

i When using restrictions for FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING, keep in mind that

- a restriction for follow-up processing must always be made for SUCCESS- and FAILURE-PROCESSING. Otherwise, it is possible that users will avoid this step.
- PREFIX of FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING must correspond,
  e.g. FILE-NAME = *EXP(XYZ.),SUCC = *EXP(’PR DSNAME(XYZ.’,')')

Example 1

Dagobert Duck wishes to create an admission profile for the following purpose:

Dussel Duck, employee at the Duck Goldmine, has his own z/OS computer. He has to transfer monthly reports on a regular basis to his boss Dagobert‘s computer, DAGODUCK, using File Transfer. The file needs to have the name MONTHLY.REPORT.GOLDMINE and is to be printed out after transfer.
The JCL statement for printing out the file MONTHLY.REPORT.GOLDMINE is completely contained in the member GOLDMOBE of the PO data set PRINT.

Since Dagobert’s admission set does not permit any “inbound” requests, he needs to give the profile privileged status (he is permitted to do this, since he is an FTAC administrator). The Goldmine computer has the security level 50. The command required to create such an admission profile is as follows:

```
FTCREPRF NAME=GOLDMOBE,                                 -
         TRANSFER-ADMISSION=MONTHLYREPORTFORTHEBOSS,    -
         PRIVILEGED=*YES,                               -
         IGNORE-MAX-LEVELS=*YES,                        -
         USER-ADM=(DONALD,XXXX,PASSWD)                  -
         TRANSFER-DIRECTION=*FROM-PARTNER,              -
         PARTNER=GOLDMINE,                              -
         FILE-NAME=MONTHLY.REPORT.GOLDMINE,             -
         SUCCESS-PROCESSING=                            -
```

```
        'ALLOC DSNAME(PRINT(MONTHLY.REPORT.GOLDMINE))',-
        FAILURE-PROCESSING=*NONE,                      -
        WRITE-MODE=*REPLACE-FILE
```

The short form of this command is:

```
FTCREPRF␣GOLDMOBE,TRANS-AD=MONTHLYREPORTFORTHEBOSS,          -
PRIV=*YES,IGN-MAX-LEV=*YES,USER-ADM=(DONALD,XXXX,PASSWD),     -
TRANS-DIR=*FROM,PART=GOLDMINE,                               -
FILE-NAME=MONTHLY.REPORT.GOLDMINE,                           -
SUCC='ALLOC DSNAME(PRINT(MONTHLY.REPORT.GOLDMINE))',FAIL=*NONE,-
WRITE=*REPL
```

File management can also be performed with this admission profile (see the specifications for the IGNORE-MAX-LEVELS operand).

Dussel Duck, who keeps the monthly report for the goldmine in his z/OS computer in the file NOTHING.BUT.LIES, can use the following openFT command to send it to the central computer DAGODUCK and print it out there:

```
/NCOPY␣TO,DAGODUCK,(NOTHING.BUT.LIES), -
       REM=*MSP(FILE=*NOT-SPECIFIED,TRANS-AD=MONTHLYREPORTFORTHEBOSS)
```

Example 2

A profile is to be created that only allows monitoring.

```
FTCREPRF MONITOR,,'ONLYFTMONITOR'  -
   ,FILE-NAME=*EXP('|*FTMONITOR ') -
   ,FT-FUN=(*TRANS-F,*FILE-PROC)
```

The openFT Monitor can be started from a Unix or Windows system using this profile with the following command:

```
ftmonitor "-po=10" FTZOS ONLYFTMONITOR
```

Alternatively, the monitoring values can be output as rows to a file (in this case ftzos_data), for instance with the following command:

```
ncopy FTZOS!"-po=10" ftzos_data ONLYFTMONITOR
```

Example 3

If you only want to use FTAC profiles for the ftexec command then you must specify a filename prefix that starts with the character string ’|ftexecsv’.

If a command or command prefix is also to be defined, you must specify it in the following form:

```
FILE-NAME=*EXP('|ftexecsv -p=command-prefix')
```

If the command string or the command prefix set in the profile for calling ftexec contains spaces, it must be enclosed in double quotes ("). Any double quotes in the command string must be entered twice.

If the entire command string is specified as a file name in the profile for ftexec, you can only specify a space (' ') as the command name when calling ftexec. The FTAC profile does not prevent a caller of ftexec from specifying further command parameters.

Example 4

You want to create a profile which can be used to run precisely one file processing command. A number of logging records are output in the example below.

```
FTCREPRF NUR1VORV,,'GetLoggingRecords'            -
,USER-ADMISSION=(DONALD,xxxx,password),           -
,FILE-NAME=*EXP('|ftexecsv -p="FTSHWLOG ,"')      -
,FT-FUN=(*TRANS-F,*FILE-PROC)
```

The following command, for example, can be used to access the profile from a remote system:

– Unix system or Windows system:

```
ftexec FTZOS 3 GetLoggingRecords
```

– BS2000 system:

```
/EXE-REM-CMD FTZOS,'3','GetLoggingRecords'
```

– z/OS system:

```
FTEXEC FTZOS,'3','GetLoggingRecords'
```

The last three logging records to be written in the z/OS are output:

```
FTSHWLOG ,3
 TYP LOGG-ID TIME     RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 2007-04-20
 C  275800   15:59:36 0000 >UNIX3 *REMOTE           UID      |FTEXECSV -P="FTSHWLOG ," 3
-B
 T  275799   15:55:55 0000 <FTBS2    UID       1L3I UID      DONALD.CH.1
 T  275798   15:55:54 0000 >FTBS2    *REMOTE        UID       2
```

6.13 FTDELKEY Delete a key pair set

Note on usage

User group: FT administrator

The command can only be specified under TSO.

Functional description

Using the DELETE-KEY-SET / FTDELKEY command, you are deleting the key pair set of a reference. The key pair consists of a private key, which is internally administered by openFT, and a public key.

Public keys are stored under:

<openft qualifier>.<inst>.SYSPKF.R<key reference>.L<key length>

Here, the first two name parts are replaced by OPENFT QUALIFIER and the name of the instance.

The key reference is a numeric designator for the version of the key pair. For each reference there are three keys with lengths of 768, 1024 and 2048 respectively.

A key pair set should only be deleted if no partner system uses the corresponding public key any longer. This means that, after creating a new key pair set using CREATE-FT-KEY-SET, the new public key should be made available to all of the partner systems in which the local system is to be authenticated.

There should always be at least one key pair set in your openFT instance, otherwise all requests will be carried out in unencrypted form.

Format

FTDELKEY
REFERENCE = <integer 1..9999999>

Operands

REFERENCE = <integer 1..9999999>
Allows selection of the key pair set to be deleted. You will find the reference in the name of the public key file (see above).

6.14 FTDELLOG Delete logging records

Note on usage

User group: FT administrator, FTAC administrator

The command can be entered under TSO.

Functional description

With FTDELLOG you can delete logging records for all login names and all record types (FT, FTAC, ADM). This function is not permitted for the ordinary user.

FTAC log records are only written when the FTAC functionality is used.

In principle, openFT can write any number of logging records (until the disk is full). The FT administrator should save the existing logging records (e.g. in hardcopy, to tape or as a file in CSV format) and at regular intervals (weekly, for example, if there is a large number of requests) and delete older logging records. This means, firstly, that logging records are retained for a long period, thereby ensuring continuous documentation, and secondly, that memory space is not occupied unnecessarily.

The logging records are saved by redirecting the output of FTSHWLOG (Displaying logging records, page 311) to a file, e.g. by executing the FTSHWLOG command as CLIST.

When deleting logging records, the PAM pages occupied by the log file are not released; the free space within the file is, however, used to store new records.

Command execution may take several minutes, depending on the size of the log file!

The default setting for the command FTDELLOG has changed. If you specify the command without parameters, the default value *PARAMETERS() is used instead of *ALL as previously, i.e. all log records are deleted that have been written up to 00:00 h of the current day. This means that the command remains downward compatible in terms of its behavior.

Format

```
FTDELLOG

SELECT = *ALL / *OWN / *PARAMETERS(...)

  *PARAMETERS(...)
   | OWNER-IDENTIFICATION = *ALL / *OWN / <name 1..8>
   |,LOGGING-DATE = *TODAY / *TOMORROW / <date 8..10>
   |,LOGGING-TIME = 00:00 / <time 1..8>
   |,RECORD-TYPE = *ALL / *PARAMETERS(...)
   |    *PARAMETERS(...)
   |     | FT = *ALL / *NONE
   |     |,FTAC = *ALL / *NONE
   |     |,ADM = *ALL / *NONE
   |,LOGGING-ID = *ALL / <alphanum-name 1..12>
```

Operands

SELECT =
Selects a group of logging records.

SELECT = *ALL
Deletes all logging records.

SELECT = *OWN
Deletes all logging records of your own ID.

SELECT = *PARAMETERS(...)

OWNER-IDENTIFICATION =
User ID whose logging records are to be deleted.

OWNER-IDENTIFICATION = *ALL
The user ID is not a selection criterion.

OWNER-IDENTIFICATION = *OWN
Logging records in the user ID are deleted.

OWNER-IDENTIFICATION = <name 1..8>
User ID whose logging records are to be deleted.

LOGGING-DATE =
Date before which the logging records are to be deleted.

LOGGING-DATE = *TODAY
If a time was specified explicitly with LOGGING-TIME, all logging records that were written before this time are deleted. If no date was specified, all logging records are deleted that were written up to midnight inclusive of the previous day.

LOGGING-DATE = *TOMORROW
All logging records that were created before the command was input are deleted.

LOGGING-DATE = <date 8..10>
Date in the format yyyy-mm-dd or yy-mm-dd, e.g. 2007-12-24 or 07-12-24 for the 24th of December, 2007. FT then deletes only those logging records that were written before the date and time specified with LOGGING-TIME and LOGGING-DATE.

LOGGING-TIME =
Logging records written up to the specified time are deleted.

LOGGING-TIME = 00:00
If a date was specified explicitly with LOGGING-DATE, FT deletes all logging records written before the specified date. If no date was specified, FT deletes all logging records that were written up to midnight inclusive of the previous day.

LOGGING-TIME = <time 1..8>
Time for the day specified with LOGGING-DATE. FT deletes all logging records written before this time. Your entry must have the format hh:mm:ss, e.g. 14:30:10.

RECORD-TYPE =
Defines the type of logging records to be deleted.

RECORD-TYPE = *ALL
The record type is not a selection criterion.

RECORD-TYPE = *PARAMETERS(...)
Type of the logging record.

> **FT = *ALL / *NONE**
> Specifies whether or not the FT logging records are to be deleted.
>
> **FTAC = *ALL / *NONE**
> Specifies whether or not FTAC logging records are to be deleted.
>
> **ADM = *ALL / *NONE**
> Specifies whether ADM log records are deleted or not.

LOGGING-ID =
Selects the logging records on the basis of the logging ID.

LOGGING-ID = *ALL
The logging ID is not a selection criterion.

LOGGING-ID = <alphanum-name 1..12>
All logging records with a logging ID smaller than or equal to the specified value are deleted.

Example

The FT administrator wants to delete all existing FT logging records. (If there is a large number of logging records, this may take several minutes!) If FTAC is not installed, logging only contains FT logging records and ADM logging records where applicable. They are deleted with the following command:

```
FTDELLOG SELECT=*PARAMETERS(LOGGING-DATE=*TOMORROW)
```

The FT administrator does not need to specify the operand OWNER-IDENTIFICATION because the standard value *ALL applies.

However, if FTAC were used then this command would delete the FT and FTAC logging records and ADM logging records where applicable because both FT=*ALL and FTAC=*ALL and ADM=*ALL are default values for RECORD-TYPE. If only the FT logging records are to be deleted, but the FTAC and ADM logging records are to be retained, then the FT administrator must extend the command:

```
FTDELLOG SELECT=*PARAMETERS(LOGGING-DATE=*TOMORROW, -
                RECORD-TYPE=*PARAMETERS(FTAC=*NONE,ADM=*NONE))
```

6.15 FTDELPRF Delete admission profile

Note on usage

User group: FTAC user and FTAC administrator

The command can be entered under TSO.

A prerequisite for using this command is the use of openFT-AC.

Functional description

With the command FTDELPRF, the FTAC user can delete all admission profiles of which he is the owner. In your role as FTAC administrator, you can delete the admission profiled of any users. You should occasionally thin out the set of profiles to ensure that there are no out-of-date admission profiles in your system that could potentially threaten the security of your system.

With SHOW-FT-PROFILE (see page 346ff), you can view the profiles and decide which ones you no longer need.

Format

```
FTDELPRF

 NAME = *ALL / <alphanum-name 1..8> / *STD
,PASSWORD = *NONE / <alphanum-name 1..8>
,SELECT-PARAMETER = *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
     |  TRANSFER-ADMISSION = *ALL / *NOT-SPECIFIED / <alphanum-name 8..32> /
     |                                     <c-string 8..32 with-low> / <x-string 15..64>
     | ,OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
```

Operands

NAME =
You can access the admission profile to be deleted using its name.

NAME = *ALL
Deletes all admission profiles. The FTAC user can delete all of his admission profiles with this operands, if he does not select a special profile with SELECT-PARAMETER.

The administrator can delete his own profiles with this entry. He can also use SELECT-PARAMETER to delete all the admission profiles of a particular user or all the admission profiles in the system.

NAME = <alphanum-name 1..8>
Deletes the admission profile with the specified name.

NAME = *STD
Deletes the default admission profile for your own user ID.

PASSWORD =
You enter the FTAC password which permits you to use FTAC commands with your user ID.

PASSWORD = <u>*NONE</u>
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
Specifies the corresponding FTAC password.
If the FTAC administrator has defined an FTAC password, then this password must be entered here, if he wishes to delete the profiles of other users.

SELECT-PARAMETER =
You can enter selection criteria for the admission profiles to be deleted.
FTAC users can address the admission profiles to be deleted using their TRANSFER ADMSSION.
FTAC administrators can address the admission profiles to be deleted using their TRANSFER ADMSSION or OWNER IDENTIFICATION.

SELECT-PARAMETER = <u>*OWN</u>
Deletes your own admission profiles.

SELECT-PARAMETER = *PARAMETERS(...)
With this structure, you can enter individual selection criteria.

> **TRANSFER-ADMISSION =**
> You can use the transfer admission of an admission profile as a selection criterion for deletion.
>
> **TRANSFER-ADMISSION = <u>*ALL</u>**
> Deletes admission profiles irrespective of the TRANSFER-ADMISSION.
>
> **TRANSFER-ADMISSION = *NOT-SPECIFIED**
> Deletes admission profiles for which no transfer admission is specified.
>
> **TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>**
> Deletes the admission profile which is accessed with this transfer admission. The alphanumeric entry is always saved in lower-case letters. The FTAC user can only enter the transfer admissions of his own admission profiles.

OWNER-IDENTIFICATION =
Deletes a specific owner’s admission profile. The FTAC user can only delete his own profiles. The FTAC administrator can also enter foreign user IDs.

OWNER-IDENTIFICATION = *OWN
Deletes your own admission profile.

OWNER-IDENTIFICATION = *ALL
Allows the FTAC administrator to delete admission profiles of all user IDs. The FTAC user is not permitted to use this entry.

OWNER-IDENTIFICATION = <alphanum-name 1..8>
The FTAC user can only specify his own user ID; the effect corresponds to *OWN. The FTAC administrator deletes the admission profiles under this user ID.

6.16 FTEXPENV Export FTAC admission profiles and sets

Note on usage

User group: FTAC administrator

openFT-AC must be installed to use this command.

The command can be entered under TSO.

Functional description

The FTAC administrator can easily “move” admission profiles and sets when a user migrates from one computer to another. The commands FTEXPENV and FTIMPENV are intended for this purpose.

This command is not available to FTAC users! Export files cannot be extended. They must be deleted and created again if necessary.

The commands only affect the currently set openFT instance. If necessary, the FTAC administrator must create them under several openFT instances.

Format

```
FTEXPENV

 TO-FILE = <filename 1..46>
,USER-IDENTIFICATION = *ALL / list-poss(100): <name 1..8>
,SELECT-PARAMETER = *ALL / *PARAMETERS(...)
   *PARAMETERS(...)
     |  PROFILE-NAME = *ALL / *NONE / list-poss(100): <alphanum-name 1..8>
     | ,ADMISSION-SET = *YES / *NO
```

Operands

TO-FILE = <filename 1..46>
Name of the file in which the admission profiles and sets are output. Temporary files may not be used.

USER-IDENTIFICATION =
The user ID whose admission profiles and sets are to be output on file.

USER-IDENTIFICATION = *ALL
The admission profiles and sets of all user IDs are to be output on file.

USER-IDENTIFICATION = list-poss(100): <name 1..8>
The admission profiles and sets of the user IDs specified are to be output on file.

SELECT-PARAMETER =
Determines whether only admission profiles, only admission sets, or both are to be output on file. For admission profiles, you can select those which are to be output.

SELECT-PARAMETER = *ALL
All admission profiles and sets associated with the user ID specified under USER-IDENTIFICATION are to be output on file.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies which of the admission profiles and sets associated with the USER-IDENTIFICATION are to be output on file.

PROFILE-NAME = *ALL
All admission profiles are output on file.

PROFILE-NAME = *NONE
No admission profiles are exported.

PROFILE-NAME = list-poss(100): <alphanum-name 1..8>
Only the profiles with the specified names (maximum 100) are output on file.

ADMISSION-SET = *YES
All admission sets are output on file.
Here, openFT only takes account of values that differ from the default (i.e. are not marked with a * in the FTSHWADS output). For all specifications that refer to the standard admission set, openFT takes over the current settings from the corresponding standard admission set when importing admission sets. The standard admission set itself is not output to file.

ADMISSION-SET = *NO
No admission sets are exported.

Example

The FTAC administrator wants to export all the admission profiles belonging to the user Billy to the external file BILLYPRF. The admission set is not to be exported:

```
FTEXPENV TO-FILE=BILLYPRF,USER-ID=BILLY,SEL=(PROF-NAME=*ALL,ADM-SET=*NO)
```

6.17 FTHELP Display information on reason codes in the logging records

Note on usage

User group: FT user and FT administrator

The command has to be entered in the TSO command mode.

Functional description

You can have the meaning of the reason codes contained in the logging records displayed by the command FTHELP (RC in the output of the command FTSHWLOG in logging records).

Format

FTHELP
<number 1..ffff>

Description

<number 1..ffff>
Stands for a four-digit reason code as it appears in the logging record. Leading zeros can be omitted during input. In an FTAC logging record, the reason code 0000 means that an FTAC admission check has permitted the request. Any other reason code indicates the reason for rejection by FTAC.
The reason code 0000 in an FT logging record indicates that file transfer has terminated successfully. All reason codes other than 0000 indicate failure.

Example

A transfer code is rejected by the local system with the following error message:

```
FTR2046 OPENFT: Local transfer admission invalid.
```

The FTAC administrator uses the command FTSHWLOG (see page 311) to display the relevant FTAC logging record. This is what the output he or she receives looks like:

```
TYP LOGG-ID TIME    RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
2007-04-24
C       77 15:19:06 3003 >JUMBO    USER001        USER001  ABC
```

The meaning of reason code 3003 can now be determined with the command FTHELP:

```
FTHELP 3003
 3003: Request rejected. Invalid password
```

Thus, the request was rejected because an invalid password was specified.

6.18 FTIMPENV Import FTAC admission profiles and sets

Note on usage

User group: FTAC administrator

openFT-AC must be installed to use this command.

This command can be entered under TSO.

Functional description

The FTAC administrator can easily “move” admission profiles and sets when a user migrates from one computer to another. The commands FTEXPENV and FTIMPENV are intended for this purpose. These commands cannot be used by the FTAC user.
All imported admission profiles will be first locked.
This can be seen in the FTSHWPRF command in the specification *LOCKED (by_import).
Privileged profiles lose their privileged status when imported. They will also be designated as private.

An admissions profile is otherwise only imported if its name does not exist on the destination ID.

If the target computer already has an admission profile with the same transfer admission and the admission profile is designated as private, both transfer admissions are locked. The transfer admission of the old profile is set to *DUPLICATED and the transfer admission of the imported profile is set to *NOT-SPECIFIED. If the already existing admission profile is designated as ”public”, then it is not locked.

Format

FTIMPENV
FROM-FILE = <filename 1..46>
,**USER-ID**ENTIFICATION = ***ALL** / list-poss(100): <name 1..8>
,**SEL**ECT-PARAMETER = ***ALL** / ***PAR**AMETERS(...)
***PAR**AMETERS(...)
│ **PROF**ILE-**NAME** = ***ALL** / ***NONE** / list-poss(100): <alphanum-name 1..8>
│ ,**AD**MISSION-**SET** = ***YES** / ***NO**
,**SECURITY** = ***STD** / ***HIGH**

Operands

FROM-FILE = <filename 1..46>
Name of the file from which the admission profiles and sets are to be imported. If the file contains invalid data or if there is an error while accessing the file, the command is rejected with the message FTC0103.

USER-IDENTIFICATION =
User ID whose admission profiles and sets are to be transferred from an export file.

USER-IDENTIFICATION = *ALL
The admission profiles and sets of all users are to be transferred.

USER-IDENTIFICATION = list-poss(100): <name 1..8>
The admission profiles and sets of the users specified (maximum 100) are to be transferred.

SELECT-PARAMETER =
Determines whether only admission profiles, only admission sets, or both are to be imported. For admission profiles, you can specify which are to be imported.

SELECT-PARAMETER = *ALL
All the admission profiles and sets associated with the user ID specified under USER-IDENTIFICATION are to be imported.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies which of the admission profiles and sets associated with the USER-IDENTIFICATION are to be imported.

PROFILE-NAME = *ALL
All admission profiles are to be imported.

PROFILE-NAME = *NONE
No admission profiles are to be imported.

PROFILE-NAME = list-poss(100): <alphanum-name 1..8>
Only the profiles specified are to be imported (maximum 100).

ADMISSION-SET = *YES
All admission sets are to be imported.

ADMISSION-SET = *NO
No admission sets are to be imported.

SECURITY =
An FTAC administrator with system administrator privilege can use this operand to control security.

SECURITY = *STD
For FTAC administrators with SU privilege:
The profile attributes are not altered when imported.

For FTAC administrators not having the SU privilege:
This operand works like the specification *HIGH, i.e. the admissions profiles are locked (locked by import) and retain the attributes USAGE=PRIVATE and PRIVILEGED = NO.

SECURITY = *HIGH
The admissions profiles are locked (locked by import) and retain the attributes USAGE=PRIVATE and PRIVILEGED=NO.

Example

The FTAC administrator wants to import all admission profiles belonging to the user Billy from the external file BILLYPRF. The admission set is not to be imported.

```
FTIMPENV FROM-FILE=BILLYPRF,USER-ID=BILLY,
         SEL=(PROF-NAME=*ALL,ADM-SET=*NO)
```

If the FTAC administrator possesses the SU privilege then the profiles can be used immediately. Otherwise, Billy must first unlock them with FTMODPRF:

```
FTMODPRF *ALL,TRANS-ADM=*OLD-ADM(VALID=*YES)
```

6.19 FTMODADS Modify admission set

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

The command can be entered under TSO.

Functional description

The FTAC user can modify the admission set for his own user ID with the FTMODADS command. The FTAC administrator also can modify the admission sets of foreign user IDs. You may access two components of the admission set:

a) You can define a password to be entered for almost all subsequent FTAC commands (except the FTSHW... commands). This prevents other users working with your user ID from entering FTAC commands.

   i It is not possible to have an FTAC password output. If an FTAC user forgets his FTAC password, only the FTAC administrator can delete or modify the password.

   ⚠ **WARNING!**
   If the FTAC administrator should assign and subsequently forget a password, the FTAC environment must be reinstalled. In this case, all admission profiles and sets are deleted!

b) FTAC users may modify the limit values for the maximum number of security levels that can be reached from their user ID (the MAX-USER-LEVELS) within the range specified by the FTAC administrator. The limit values defined by the FTAC administrator (MAX-ADM-LEVELS) cannot, however, be overridden by the FTAC user. They can simply reduce the limit values since, in the case of FT requests, FTAC performs the admission check on the basis of the smallest value in the admission set. The MAX-USER-LEVELS are only effective if they are lower, i.e. more restrictive, than the MAX-ADM-LEVELS.

   FTAC administrators assign a maximum security level for each of the six basic functions. The user ID associated with the admission set can then use this function with all partner systems with this security level or lower. The owner of the admission set may only increase the degree of restriction.

In addition, the FTAC administrator can delete an admission set from the admission file by entering the default admission set for the user ID in question (MAX-LEVELS=*STD). This is also possible with user IDs which have already been deleted!

Format

```
FTMODADS

 USER-IDENTIFICATION = *OWN / *STD / <name 1..8>
,PASSWORD = *NONE / <alphanum-name 1..8>
,SELECT-PARAMETER = *ALL
,NEW-PASSWORD = *OLD / *NONE / <alphanum-name 1..8>
,PRIVILEGED = *UNCHANGED
,MAX-LEVELS = *UNCHANGED / *STD / <integer 0...100> / *PARAMETERS(...)
    *PARAMETERS(...)
      |  OUTBOUND-SEND = *UNCHANGED / *STD / <integer 0...100>
      | ,OUTBOUND-RECEIVE = *UNCHANGED / *STD / <integer 0...100>
      | ,INBOUND-SEND = *UNCHANGED / *STD / <integer 0...100>
      | ,INBOUND-RECEIVE = *UNCHANGED / *STD / <integer 0...100>
      | ,INBOUND-PROCESSING = *UNCHANGED / *STD / <integer 0...100>
      | ,INBOUND-MANAGEMENT = *UNCHANGED / *STD / <integer 0...100>
```

Operands

USER-IDENTIFICATION =
User ID whose admission set is to be modified.

USER-IDENTIFICATION = *OWN
The admission set for the user ID which you are currently using is to be modified.

USER-IDENTIFICATION = *STD
The default admission set is to be modified. Only the FTAC administrator can make this entry.

USER-IDENTIFICATION = <name 1..8>
The admission set for this user ID is to be modified. The FTAC user can only enter his own user ID here.
The FTAC administrator can enter any user ID here.

PASSWORD =
FTAC password which authorizes you to use FTAC commands, if such a password was defined in your admission set. An FTAC password is set with the operand NEW-PASSWORD.

PASSWORD = *NONE
No FTAC password is required for this admission set.

PASSWORD = <alphanum-name 1..8>
This password authorizes this user to use FTAC commands.

SELECT-PARAMETER = *ALL
In later openFT-AC versions it will be possible to specify additional selection criteria here.

NEW-PASSWORD =
Changes the FTAC password. If such an FTAC password has already been set, it must be used for almost all FTAC commands on the user ID for this admission set (except: the FTSHW... commands). This is done using the parameter PASSWORD in the respective commands.

NEW-PASSWORD = *OLD
The FTAC password remains unchanged.

NEW-PASSWORD = *NONE
No FTAC password is required for the user ID associated with this admission set.

NEW-PASSWORD = <alphanum-name 1..8>
Specification of the new FTAC password.

PRIVILEGED = *UNCHANGED
This parameter is only supported for reasons of compatibility. Authorization of the FTAC administrator is now only possible via the FTACADM member in the openFT parameter library.

MAX-LEVELS =
You set which security level(s) you can access, with which basic functions, from the user ID of this admission set. Either you can set one security level for all basic functions or different security levels for each basic function.
The MAX-USER-LEVELS for this admission set are set by the FTAC user; the MAX-ADM-LEVELS are set by the FTAC administrator.
FTAC runs authorization checks on the basis of the lowest specified security level. FTAC users may reduce but not increase the values specified for them by the FTAC administrator, see example to FTSHWADS.

MAX-LEVELS = *UNCHANGED
The security levels set in this admission set are to remain unchanged.

MAX-LEVELS = *STD
For this admission set, the values of the default admission set are valid. The admission set is deleted from the admission file. This is possible if the user ID has already been deleted.

MAX-LEVELS = <integer 0...100>
You can set a maximum security level for all six basic functions. The value 0 means that no file transfer is possible on this user ID until further notice (until the admission set is modified again).

MAX-LEVELS = *PARAMETERS(...)
You can set a maximum security level for each of the basic functions.

OUTBOUND-SEND =
Sets the maximum security level for the basic function “outbound send”. The owner of the admission set can send files to all partner systems whose security level has this value or lower.

OUTBOUND-SEND = *UNCHANGED
The value for OUTBOUND-SEND remains unchanged.

OUTBOUND-SEND = *STD
For OUTBOUND-SEND, the value from the default admission set is used.

OUTBOUND-SEND = <integer 0..100>
For OUTBOUND-SEND, this maximum security level is entered in the admission set.

OUTBOUND-RECEIVE =
Sets the maximum security level for the basic function “outbound receive”. The owner of the admission set can receive files from all partner systems whose security level has this value or lower.

OUTBOUND-RECEIVE = *UNCHANGED
The value for OUTBOUND-RECEIVE remains unchanged.

OUTBOUND-RECEIVE = *STD
For OUTBOUND-RECEIVE, the value from the default admission set is used.

OUTBOUND-RECEIVE = <integer 0..100>
For OUTBOUND-RECEIVE, this maximum security level is entered in the admission set.

INBOUND-SEND =
Sets the maximum security level for the basic function “inbound send”. All partner systems with this security level or lower can request files from the owner of the admission set.

INBOUND-SEND = *UNCHANGED
The value for INBOUND-SEND remains unchanged.

INBOUND-SEND = *STD
For INBOUND-SEND, the value from the default admission set is used.

INBOUND-SEND = <integer 0..100>
For INBOUND-SEND, this maximum security level is entered in the admission set.

INBOUND-RECEIVE =
Sets the maximum security level for the basic function “inbound receive”. All partner systems with this security level or lower may send files to the owner of the admission set.

INBOUND-RECEIVE = *UNCHANGED
The value for INBOUND-RECEIVE remains unchanged.

INBOUND-RECEIVE = *STD
For INBOUND-RECEIVE, the value from the default admission set is used.

INBOUND-RECEIVE = <integer 0..100>
For INBOUND-RECEIVE, this maximum security level is entered in the admission set.

INBOUND-PROCESSING =
Sets the maximum security level for the basic function “inbound processing”. All partner systems which have this security level or lower may include follow-up processing in their system as part of an FT request.

INBOUND-PROCESSING = *UNCHANGED
The value for INBOUND-PROCESSING remains unchanged.

INBOUND-PROCESSING = *STD
For INBOUND-PROCESSING, the value from the default admission set is used.

INBOUND-PROCESSING = <integer 0..100>
For INBOUND-PROCESSING, this maximum security level is entered in the admission set.

INBOUND-MANAGEMENT =
Sets the maximum security level for the basic function “inbound file management”. All partner systems with this security level or lower may include the modification of file attributes and the querying of directories as part of their FT request.

INBOUND-MANAGEMENT = *UNCHANGED
The value for INBOUND-MANAGEMENT remains unchanged.

INBOUND-MANAGEMENT = *STD
For INBOUND-MANAGEMENT, the value from the default admission set is used.

INBOUND-MANAGEMENT = <integer 0..100>
For INBOUND-MANAGEMENT, this maximum security level is entered in the admission set.

Example

Dagobert Duck, the FTAC administrator of the Duck Bank, wishes set up the admission set for his employee Donald, such that Donald

– can send files to partner systems with the security level of 10 or lower (basic function “outbound send”),

– can request files from partner systems with the security level of 10 or lower (basic function “outbound receive”).

He wants all partner systems to be able send files to and request files from the user ID DONALD, so he sets the security level for INBOUND-SEND and INBOUND-RECEIVE to 100.

Dagobert does not wish to permit follow-up processing to be initiated from external partners, since he is too stingy to want to make his resources available to others. Therefore, he sets INBOUND-PROCESSING and INBOUND-FILEMANAGEMENT at 0. Since these values are set in the default admission set for the Duck Bank, these specifications are used for *STD. No FTAC password is defined.

The long form of the required command is as follows:

```
FTMODADS USER-IDENTIFICATION=DONALD,    -
         MAX-LEVELS=(OUTBOUND-SEND=10,  -
         OUTBOUND-RECEIVE=10,           -
         INBOUND-SEND=100,              -
         INBOUND-RECEIVE=100,           -
         INBOUND-PROCESSING=*STD,       -
         INBOUND-MANAGEMENT=*STD)
```

A possible short form of this command would be:

```
FTMODADS DONALD,MAX-LEV=(10,10,100,100,*STD,*STD)
```

6.20 FTMODOPT Modify operating parameters

Note on usage

User group: FT administrator

Functional description

The FTMODOPT command is used to modify one or more operating parameters of the local system. The relationships between the different operating parameters are explained in section “Optimizing the operating parameters” on page 99.

After first installation of the instance the authentication check is deactivated. Using FTMODOPT it is possible to activate and (if necessary) deactivate the authentication check (PARTNER-CHECK parameter). This parameter can be individually configured for each partner system using FTMODPTN. The expanded sender checking is only valid for partner systems that are not authenticated in the current openFT instance.

If, after installation, the security level at partners was not set with FTADDPTN or FTMODPTN then the following automatic mechanism applies (SECURITY-LEVEL=*BY-PARTNER-ATTRIBUTES):

- Authenticated partners are assigned the security level 10,
- Partners, known in the transport system are assigned the security level 90,
- Partners accessed via their IP address are assigned the security level 100.

Any unspecified operating parameters remain unchanged. The current operating parameters can be queried at any time using the FTSHWOPT command   (see page 339).

The FTMODOPT command is also used to switch the FT trace, and console traps and ADM traps on and off and to control FT logging and monitoring.

Format

(part 1 of 2)

```
FTMODOPT

 PROCESS-LIMIT = *UNCHANGED / <integer 1..32>
,CONNECTION-LIMIT = *UNCHANGED / <integer 1..99>
,REQUEST-WAIT-LEVEL = *UNCHANGED
,PACING = *UNCHANGED
,TRANSPORT-UNIT-SIZE = *UNCHANGED / <integer 512..65535>
,SECURITY-LEVEL = *UNCHANGED / *BY-PARTNER-ATTRIBUTES / <integer 1..100>
,PARTNER-CHECK = *UNCHANGED / *STD / *TRANSPORT-ADDRESS
,TRACE = *UNCHANGED / *ON / *OFF / *CHANGE-FILES / *PARAMETERS(...)
   *PARAMETERS(...)
     |  SWITCH = *UNCHANGED / *ON / *OFF / *CHANGE-FILES
     | ,PARTNER-SELECTION = *UNCHANGED / *ALL / *NONE / list-poss(3): *OPENFT / *FTP/ *ADM
     | ,REQUEST-SELECTION = *UNCHANGED / *ALL / list-poss(2): *ONLY-SYNC / *ONLY-ASYNC /
     |                                *ONLY-LOCAL / *ONLY-REMOTE
     | ,OPTIONS = *UNCHANGED / *NONE / list-poss(1): *NO-BULK-DATA
,LOGGING = *UNCHANGED / *SELECT(...)
   *SELECT(...)
     |  TRANSFER-FILE = *UNCHANGED / *OFF / *ON / *FAILURE
     | ,FTAC = *UNCHANGED / *ON / *REJECTED / *MODIFICATIONS
     | ,ADM = *UNCHANGED / *OFF / *ON / *FAILURE/ *MODIFICATIONS
,MAX-INBOUND-REQUEST = *UNCHANGED
,REQUEST-LIMIT = *UNCHANGED / <integer 2..32000>
,MAX-REQUEST-LIFETIME = *UNCHANGED / *UNLIMITED / <integer 1..400>
,SNMP-TRAPS = *UNCHANGED / *NONE
,CONSOLE-TRAPS =  *UNCHANGED *ALL / *NONE / *PARAMETERS(...)
   *PARAMETERS(...)
     |  SUBSYSTEM-STATE = *UNCHANGED / *OFF / *ON
     | ,FT-STATE = *UNCHANGED / *OFF / *ON
     | ,PARTNER-STATE = *UNCHANGED / *OFF / *ON
     | ,PARTNER-UNREACHABLE = *UNCHANGED / *OFF / *ON
     | ,REQUEST-QUEUE-STATE = *UNCHANGED / *OFF / *ON
     | ,TRANSFER-SUCCESS = *UNCHANGED / *OFF / *ON
     | ,TRANSFER-FAILURE = *UNCHANGED / *OFF / *ON
```

(part 2 of 2)

```
,HOST-NAME = *UNCHANGED
,IDENTIFICATION = *UNCHANGED / <c-string 1..64 with-low> / <composed-name 1..64>
,KEY-LENGTH = *UNCHANGED / 0 / 768 / 1024 / 2048
,CODED-CHARACTER-SET = *UNCHANGED / <alphanum-name 1..8>
,OPENFT-APPLICATION = *UNCHANGED / *STD / <text 1..24>
,OPENFT-STD = *UNCHANGED / *STD /  <integer 1..65535>
,FTAM-APPLICATION = *UNCHANGED
,FTP-PORT = *UNCHANGED / *NONE / *STD / <integer 1..65535>
,DYNAMIC-PARTNERS = *UNCHANGED / *OFF / *ON
,ADM-PORT = *UNCHANGED / *STD / <integer 1..65535>
,ACTIVE-APPLICATIONS = *UNCHANGED / *ALL / *NONE / list-poss(3): *OPENFT / *ADM / *FTP
,ADM-CONNECTION-LIMIT = *UNCHANGED / <integer 1..99>
,MONITORING  = *UNCHANGED / *ON / *OFF / *PARAMETERS(...)
   *PARAMETERS(...)
   |  SWITCH = *UNCHANGED / *ON / *OFF
   |  ,PARTNER-SELECTION = *UNCHANGED / *ALL / *NONE / list-poss(2): *OPENFT / *FTP
   |  ,REQUEST-SELECTION = *UNCHANGED / *ALL / list-poss(2): *ONLY-SYNC / *ONLY-ASYNC /
   |                              *ONLY-LOCAL / *ONLY-REMOTE
,ADM-TRAPS = *UNCHANGED / *NONE / *PARAMETERS(...)
   *PARAMETERS(...)
   |  DESTINATION = *UNCHANGED / *NONE / *PARAMETERS(...)
   |     *PARAMETERS(...)
   |     |  PARTNER = *UNCHANGED / <text 1..200 with-low>
   |     |  ,TRANSFER-ADMISSION = *UNCHANGED / <alphanum-name 8..32> /
   |     |                        <c-string 8..32 with-low> / <x-string15..64> /
   |  ,SELECTION = *UNCHANGED / *ALL/ *NONE / *PARAMETERS(...)
   |     *PARAMETERS(...)
   |     |  ,FT-STATE = *UNCHANGED / *OFF / *ON
   |     |  ,PARTNER-STATE = *UNCHANGED / *OFF / *ON
   |     |  ,PARTNER-UNREACHABLE = *UNCHANGED / *OFF / *ON
   |     |  ,REQUEST-QUEUE-STATE = *UNCHANGED / *OFF / *ON
   |     |  ,TRANSFER-SUCCESS = *UNCHANGED / *OFF / *ON
   |     |  ,TRANSFER-FAILURE = *UNCHANGED / *OFF / *ON
```

Operands

PROCESS-LIMIT =
Maximum number of tasks that can be reserved simultaneously for the execution of file transfer requests.

PROCESS-LIMIT = *UNCHANGED
PROCESS-LIMIT is not changed, default value.

PROCESS-LIMIT = <integer 1..32>
PROCESS-LIMIT can have any value between 1 and 32.
The default value is 2.

CONNECTION-LIMIT =
Maximum number of transport connections that can be reserved for the execution of FT requests. This limit does not include file management requests and synchronous requests.

The maximum number of transport connections cannot be exceeded, not even if there are many high-priority file transfer requests to be executed. Since only one request can be processed at a time per transport connection, CONNECTION-LIMIT is also the maximum number of requests which a system can process simultaneously. One third of the transport connections defined by CONNECTION-LIMIT are reserved for requests from the remote system, and another third for requests submitted in the local system. The remaining third are available for both local and remote requests. This prevents locally submitted requests from blocking the system against requests from remote systems. If CONNECTION-LIMIT is less than 3, no transport connections are reserved.

CONNECTION-LIMIT = *UNCHANGED
The CONNECTION-LIMIT value is not changed, default value.

CONNECTION-LIMIT = <integer 1..99>
CONNECTION-LIMIT can have any value between 1 and 99.
Default value after installation: 16

REQUEST-WAIT-LEVEL = *UNCHANGED
The value for REQUEST-WAIT-LEVEL is unchanged.

PACING = *UNCHANGED
PACING is now only supported for reasons of compatibility.

TRANSPORT-UNIT-SIZE =
Maximum size of a transport unit in bytes.

TRANSPORT-UNIT-SIZE = *UNCHANGED
The current value size of a transport unit in bytes is unchanged.

TRANSPORT-UNIT-SIZE = <integer 512..65535>
TRANSPORT-UNIT-SIZE can assume any value between 512 and 65535.
The default value: 65535 bytes after installation.
It is recommended that you use this default value.

SECURITY-LEVEL =
This parameter need only be specified when FTAC functionality is used. An important part of the access protection functions provided by this product is based on the allocation of a security level to each partner. These security levels are designated using integers. The FT administrator can define a global value. This security level applies to all partner systems that are not explicitly assigned their own security levels when entered in the network description.

SECURITY-LEVEL = *UNCHANGED
The security level is unchanged.

SECURITY-LEVEL = *BY-PARTNER-ATTRIBUTES
If you set the operand to *BY-PARTNER-ATTRIBUTES then the security level is defined automatically. This setting assigns partners that are authenticated by openFT the security level 10. Partners that are known in the transport system are assigned the security level 90. All other partners are assigned security level 100. This setting is the default as of openFT V10.

SECURITY-LEVEL = <integer 1..100>
SECURITY-LEVEL can assume any value between 1 and 100. The default value is 1. If FTAC functionality is to be used, remember that 1 is the lowest level of security, offering the least protection. This is sufficient if you do not wish to further differentiate your remote systems; otherwise, a higher value should be defined. The allocation of different security levels is particularly meaningful if the authentication check is activated.

PARTNER-CHECK =
Activates the extended authentication check. When using expanded sender checking, not only the partner identification is checked, but also the transport address. PARTNER-CHECK only affects openFT partners that are not authenticated in the current openFT instance (see section “Authentication” on page 113).

The globally set expanded sender checking can be modified for specific partners. See the operand PARTNER-CHECK for the ADD-FT-PARTNER and MODIFY-FT-PARTNER commands.

PARTNER-CHECK = *UNCHANGED
The existing value is retained.

PARTNER-CHECK = *STD
If dynamic partners are prohibited (DYNAMIC-PARTNERS=*OFF), a check is performed to determine whether the partner is entered in the partner list as a partner system, and only then will the file transfer be allowed. The default value after installation is *STD. In the FTSHWOPT command then PARTNER-CHECK = *STD is output.

If dynamic partners are permitted (DYNAMIC-PARTNERS=*ON), transfers are also permitted from partners that are accessed via their address and/or are not entered in the partner list.

PARTNER-CHECK = *TRANSPORT-ADDRESS
Extended authentication check. In addition to checking whether the partner is entered in its own partner list as a partner system, it is checked whether the transport address under which the partner logs on matches the transport address entered in the partner list for the partner system. In the FTSHWOPT command then PARTNER-CHECK = *YES is output. If dynamic partners are permitted then this setting has no significance.

TRACE =
Switches FT trace functions on.

TRACE = <u>*UNCHANGED</u>
The existing FT trace functions remain unchanged.

TRACE = *ON
Switches the FT trace functions on.

TRACE = *OFF
Switches the FT trace functions off.

TRACE = *CHANGE-FILES
Switches to a new trace file. This allows a continuous trace to be created across several files to prevent a single trace file from becoming too large.

TRACE = *PARAMETERS(...)
Option that is to be applied when writing the trace.

> **SWITCH =**
> Deactivates the FT trace functions for the selected partners.
>
> **SWITCH = <u>*UNCHANGED</u>**
> The previous value is unchanged.
>
> **SWITCH = *ON**
> Activates the FT trace functions.
>
> **SWITCH = *OFF**
> Deactivates the FT trace functions.
>
> **SWITCH = *CHANGE-FILES**
> Switches to a new trace file. This allows a continuous trace to be created across several files to prevent a single trace file from becoming too large.
>
> **PARTNER-SELECTION =**
> Selects the partners that are to be traced. The selection made here can be modified with the TRACE operand of the FTMODPTN command.

PARTNER-SELECTION = *UNCHANGED
The previous value is unchanged.

PARTNER-SELECTION = *ALL
All the partners are selected for tracing.

PARTNER-SELECTION = *NONE
No partner is selected for tracing. Only those partners are traced which have been selected for tracing with the TRACE operand of the FTMODPTN command.

PARTNER-SELECTION = *OPENFT
All partners which are addressed via the openFT protocol are selected for tracing.

PARTNER-SELECTION = *FTP
All partners which are addressed via the FTP protocol are selected for tracing.

PARTNER-SELECTION = *ADM
All administration partners are selected for monitoring.

REQUEST-SELECTION =
Selects the request types that are to be traced.

REQUEST-SELECTION = *UNCHANGED
The previous value is unchanged.

REQUEST-SELECTION = *ALL
All the requests are selected for tracing.

REQUEST-SELECTION = *ONLY-SYNC
All synchronous requests are selected for tracing. Synchronous requests are always issued locally.

REQUEST-SELECTION = *ONLY-ASYNC
All asynchronous requests are selected for tracing. Requests issued remotely are always regarded as asynchronous.

REQUEST-SELECTION = *ONLY-LOCAL
All locally submitted requests are selected for tracing.

REQUEST-SELECTION = *ONLY-REMOTE
All remotely submitted requests are selected for tracing.

OPTIONS =
Controls the options for the trace functions.

OPTIONS = *UNCHANGED
The previous value is unchanged.

OPTIONS = *NONE
No options are selected for the trace functions.

OPTIONS = *NO-BULK-DATA
If file contents (bulk data) are transferred with a protocol element and multiple trace records with the same protocol element occur in succession then only the first of these records is written to the trace file. This reduces the volume of the trace file.

LOGGING =
Switches the logging functions.

LOGGING = *UNCHANGED
The existing logging functions remain unchanged.

LOGGING = *SELECT(...)
Controls logging for FT, FTAC and administration functions.

TRANSFER-FILE = *UNCHANGED
The previous settings for FT logging remain unchanged.

TRANSFER-FILE = *OFF
Switches the FT logging functions off.

TRANSFER-FILE = *ON
Switches the FT logging functions on.

TRANSFER-FILE = *FAILURE
Only failed requests are written to the logging file.

FTAC = *UNCHANGED
The previous settings for FTAC logging remain unchanged.

FTAC = *ON
Switches the FTAC logging functions on.

FTAC = *REJECTED
All requests rejected by FTAC are logged.

FTAC = *MODIFICATIONS
All modifying requests are logged.

ADM = *UNCHANGED
The previous settings for administration logging remain unchanged.

ADM = *OFF
Deactivates administration logging.

ADM = *ON
Activates administration logging.

ADM = *FAILURE
Only failed administration requests are written to the log file.

ADM = *MODIFICATIONS
Only administration requests that modify data are written to the log file.

MAX-INBOUND-REQUEST = *UNCHANGED
MAX-INBOUND-REQUEST is now only supported for reasons of compatibility.

REQUEST-LIMIT =
Changes the number of requests which can be saved in the request queue.
It generally makes no sense to reduce the size of the request queue.
If you increase the size, this only takes effect after openFT has subsequently been stopped, the batch job (or started task) has been terminated and then the batch job (or started task) and openFT have been restarted.

REQUEST-LIMIT = *UNCHANGED
The previous value remains unchanged.

REQUEST-LIMIT = <integer 2..32000>
The maximum number of requests which can be saved in the request queue is changed to the value specified.

MAX-REQUEST-LIFETIME =
Limits the lifetime of FT requests in the request file. The maximum lifetime applies to inbound and outbound requests and is specified in days. The default value when a new request file is generated is 30 days.
This parameter also affects requests to which a cancel time was explicitly assigned on request allocation (in z/OS, with the CANCEL parameter in the NCOPY command). The request is aborted as soon as either the lifetime expires or the cancel time is reached, depending on which occurs first.

MAX-REQUEST-LIFETIME = *UNCHANGED
The previous value remains unchanged.

MAX-REQUEST-LIFETIME = *UNLIMITED
The lifetime of FT requests is unlimited.

MAX-REQUEST-LIFETIME = <integer 1..400>
The maximum lifetime for FT requests may have a value of between 1 and 400 days.

SNMP-TRAPS = *UNCHANGED / *NONE
SNMP traps are not supported by openFT in z/OS systems. The parameter is only present to ensure compatibility with other openFT products and should not be changed.

CONSOLE-TRAPS =
Activates or deactivates console traps.
By default, these trap messages are logged as asynchronous messages.
They can therefore cause storage problems on systems with high request volumes.
By default, the output of console traps is activated.

CONSOLE-TRAPS = *UNCHANGED
The previous value is unchanged.

CONSOLE-TRAPS = *ALL
openFT outputs the FTR03XX console messages as asynchronous messages. Like the other asynchronous messages, the console messages are written to the job log. Depending on the ROUTCDE parameter in the PARM member of the parameter library, these messages may also optionally be written to a console.

CONSOLE-TRAPS = *NONE
The FTR03XX console messages are not output.

CONSOLE-TRAPS = *PARAMETERS(...)
Explicit specification of the events for which FTR03XX console messages are output.

SUBSYSTEM-STATE =
Controls the output of console messages concerning the status of the openFT subsystems.

SUBSYSTEM-STATE = <u>*UNCHANGED</u>
The previous value is unchanged.

SUBSYSTEM-STATE = *OFF
No console messages concerning the status of the openFT subsystem are output.

SUBSYSTEM-STATE = *ON
Console messages concerning the status of the openFT subsystem are output.

FT-STATE =
Controls the output of console messages concerning the status of the openFT control process.

FT-STATE = <u>*UNCHANGED</u>
The previous value is unchanged.

FT-STATE = *OFF
No console messages concerning the status of the openFT control process are output.

FT-STATE = *ON
Console messages concerning the status of the openFT control process are output.

PARTNER-STATE =
Controls the output of console messages concerning the status of the partner systems.

PARTNER-STATE = <u>*UNCHANGED</u>
The previous value is unchanged.

PARTNER-STATE = *OFF
No console messages concerning the status of partner systems are output.

PARTNER-STATE = *ON
Console messages concerning the status of partner systems are output.

PARTNER-UNREACHABLE =
Controls the output of console messages if partner systems cannot be accessed.

PARTNER-UNREACHABLE = *UNCHANGED
The previous value is unchanged.

PARTNER-UNREACHABLE = *OFF
No console messages are output if partner systems cannot be accessed.

PARTNER-UNREACHABLE = *ON
Console messages are output if partner systems cannot be accessed.

REQUEST-QUEUE-STATE =
Controls the output of console messages concerning the status of the request queue.

REQUEST-QUEUE-STATE = *UNCHANGED
The previous value is unchanged.

REQUEST-QUEUE-STATE = *OFF
No console messages concerning the status of the request queue are output.

REQUEST-QUEUE-STATE = *ON
Console messages concerning the status of the request queue are output.

TRANSFER-SUCCESS =
Controls the output of console messages when a request is terminated successfully.

TRANSFER-SUCCESS = *UNCHANGED
The previous value is unchanged.

TRANSFER-SUCCESS = *OFF
No console messages are output if a request is terminated successfully.

TRANSFER-SUCCESS = *ON
Console messages are output if a request is terminated successfully.

TRANSFER-FAILURE =
Controls the output of console messages when a request fails.

TRANSFER-FAILURE = *UNCHANGED
The previous value is unchanged.

TRANSFER-FAILURE = *OFF
No console messages are output if a request fails.

TRANSFER-FAILURE = *ON
Console messages are output if a request fails.

HOST-NAME = *UNCHANGED
The host name remains unchanged. This parameter is supported for reasons of compatibility only.

IDENTIFICATION =
Local instance ID of your openFT instance. With the aid of this instance ID, openFT partners as of V8.1 manage the resources for your openFT instance.

The instance ID must be unique, network-wide and must not be case-sensitive. An instance ID may consist of alphanumeric characters or special characters and may have a maximum length of 64 characters. It is advisable only to use the special characters “.”, “-”, “:” or “%”. The initial character must be alphanumeric or the special character “%”. The character “%” may only be used as an initial character. The character “.” must be followed by an alphanumeric character. For further details on assigning instance identifications, see section “Instance identification” on page 114.

When an instance is installed for the first time, the VTAM name of the real host under which their instance operates is entered as the default value. If another identification is to be used for operation then this must be configured with FTMODOPT.

IDENTIFICATION = *UNCHANGED
The instance ID remains unchanged.

IDENTIFICATION = <c-string 1..64 with-low> / <composed-name 1..64>
The instance ID is set to this value.

KEY-LENGTH =
Length of the RSA key used for encryption. This key is used only to encrypt the AES key which is agreed on between the partners (or the DES key up to and including openFT V7.0). openFT uses the AES key to encrypt the request description data and possibly also the file contents.

KEY-LENGTH = *UNCHANGED
The previous value is unchanged.

KEY-LENGTH = 0
Explicitly disables encryption.

KEY-LENGTH = 768 / 1024 / 2048
Key length in bits.

CODED-CHARACTER-SET =
Coding (character set) to be used when reading or writing a local text file during a transfer request. You can explicitly assign a different character set to a file in the transfer request or using the FT parameter library (see page 80).

CODED-CHARACTER-SET = *UNCHANGED
The character set used by default to read or write the local file is the character set that is set in the system (default: IBM1047).

CODED-CHARACTER-SET = <alphanum-name 1..8>
Name of the character set used by default to read or write a local text file. The character set must be known in the local system. openFT provides a range of character sets, see page 103.
This specification is only relevant for requests to openFT partners.

OPENFT-APPLICATION =
Specifies a port number and/or a transport selector for the local openFT server.
Use this function carefully as it will be more difficult for the openFT partners to address the local system if the port number or transport selector differ from the default values!

OPENFT-APPLICATION = *UNCHANGED
The previous value is unchanged.

OPENFT-APPLICATION = *STD
The port number and transport selector are set to the default value, i.e.:
Port number: 1100
Transport selector: $FJAM in EBCDIC code, followed by three spaces.

OPENFT-APPLICATION = <text 1..24>
Valid port number and/or a transport selector in the form [<port number>].[tsel].

OPENFT-STD =
Port number other than the default when addressing openFT partners via their host names.
Use this function carefully, as changing the port number from the default value means that it will no longer be possible to reach openFT partners which use the default port number and are addressed via the host name!

OPENFT-STD = *UNCHANGED
The previous value is unchanged.

OPENFT-STD = *STD
The port number is set to the default value 1100.

OPENFT-STD = <integer 1..65535>
Valid port number.

FTAM-APPLICATION = *UNCHANGED
This value is not relevant for z/OS systems and cannot be changed.

FTP-PORT =
This option allows you to specify the port number used by FTP.

FTP-PORT = *UNCHANGED
The previous value is unchanged.

FTP-PORT = *NONE
No port number is defined. The FTP server is deactivated, i.e. it cannot accept any inbound FTP requests. This setting is only supported for reasons of compatibility. Instead, you should use the operand ACTIVE-APPLICATIONS to activate and deactivate the inbound FTP server.

FTP-PORT = *STD
The port number is set to the default value 21.

FTP-PORT = <integer 1..65535>
Valid port number.

DYNAMIC-PARTNERS =
Specifies whether dynamic partners are permitted.

DYNAMIC-PARTNERS = *UNCHANGED
The previous value is unchanged.

DYNAMIC-PARTNERS = *OFF
Dynamic partners are not permitted. This means that it is only possible to access partner systems which are entered in the partner list and are addressed via the partner name. Transfer requests with partners which are not entered in the partner list are not permitted.

If the trace function is already switched on, the command FTMODOPT TRACE=*ON(...) has no effect; the trace scope cannot be modified for a trace run that is already underway.

DYNAMIC-PARTNERS = *ON
Dynamic partners are permitted. This means that partner systems can also be accessed via their addresses and that transfer requests are also permitted with partners which are not entered in the partner list.

ADM-PORT =
This option allows you to specify the port number used for remote administration.

ADM-PORT = *UNCHANGED
The previous value is unchanged**.**

ADM-PORT = *STD
The port number is set to the default value 11000.

ADM-PORT = <integer 1..65535>
Specifies a valid port number

ACTIVE-APPLICATIONS=
This option allows you to activate or deactivate the asynchronous inbound server.

ACTIVE-APPLICATIONS = *UNCHANGED
The previous value is unchanged**.**

ACTIVE-APPLICATIONS = *ALL
The asynchronous inbound servers for the openFT, ADM and FTP protocols are activated.

ACTIVE-APPLICATIONS = *NONE
The asynchronous inbound servers for the openFT, ADM and FTP protocols are deactivated.

ACTIVE-APPLICATIONS = list-poss(3): *OPENFT / *ADM / *FTP
You can activate the asynchronous inbound servers for specific protocols (openFT, ADM, and/or FTP), by specifying a comma-delimited list of one or more asynchronous inbound servers listed.
The asynchronous inbound servers for the protocol types that are not in the list are then automatically deactivated.

ACTIVE-APPLICATIONS = *OPENFT
Activates the asynchronous inbound server for requests via the openFT protocol.

ACTIVE-APPLICATIONS = *ADM
Activates the asynchronous inbound server for administration requests.

ACTIVE-APPLICATIONS = *FTP
Activates the asynchronous inbound server for requests via the FTP protocol.

ADM-CONNECTION-LIMIT =
This allows you to specify the maximum number of connections for remote administration.

ADM-CONNECTION-LIMIT = <u>*UNCHANGED</u>
The previous value is unchanged.

ADM-CONNECTION-LIMIT = <integer 1..99>
You can enter a value between 1 and 99 here.
The default value after installation is 8.

MONITORING =
Activates or deactivates the monitoring functions.

MONITORING = <u>*UNCHANGED</u>
The monitoring settings remain unchanged.

MONITORING = *ON
Activates monitoring without changing the filter.

MONITORING = *OFF
Deactivates monitoring.

MONITORING = *PARAMETERS(...)
Selects the options that are to be applied to monitoring.

> **SWITCH =**
> Activates or deactivates monitoring for the selected partners.
>
> **SWITCH = <u>*UNCHANGED</u>**
> The previous value is unchanged.

SWITCH = *ON
Activates monitoring.

SWITCH = *OFF
Deactivates monitoring.

PARTNER-SELECTION =
Selects the partners that are to be monitored.

PARTNER-SELECTION = *UNCHANGED
The previous value is unchanged.

PARTNER-SELECTION = *ALL
All the partners are selected for monitoring.

PARTNER-SELECTION = *NONE
No partner is selected for monitoring. In this event, only certain monitoring data values are populated. See the section "Description of the monitoring values" on page 330.

PARTNER-SELECTION = *OPENFT
All partners which are addressed via the openFT protocol are selected for monitoring.

PARTNER-SELECTION = *FTP
All partners which are addressed via the FTP protocol are selected for monitoring.

REQUEST-SELECTION =
Selects the request types for which monitoring data is to be collected.

REQUEST-SELECTION = *UNCHANGED
The previous value is unchanged.

REQUEST-SELECTION = *ALL
All requests are selected for monitoring.

REQUEST-SELECTION = *ONLY-SYNC
All synchronous requests are selected for monitoring. Synchronous requests are always issued locally.

REQUEST-SELECTION = *ONLY-ASYNC
All asynchronous requests are selected for monitoring. Requests issued remotely are always regarded as asynchronous.

REQUEST-SELECTION = *ONLY-LOCAL
All locally submitted requests are selected for monitoring.

REQUEST-SELECTION = *ONLY-REMOTE
All remotely submitted requests are selected for monitoring.

ADM-TRAPS =
Specifies the settings for the ADM trap server and the ADM traps.

ADM-TRAPS = *UNCHANGED
The previous settings remain unchanged.

ADM-TRAPS = *NONE
The ADM trap server is removed from the list, the FTAC transfer admission is deleted and all ADM traps are deactivated.

ADM-TRAPS = *PARAMETERS(...)
Changes the name of the destination, i.e. the ADM trap server and the associated FTAC transfer admission and activates or deactivates selected ADM traps.

DESTINATION =
Here you specify the name of the destination or the ADM trap server together with the corresponding FTAC transfer admission.

DESTINATION = *UNCHANGED
The name of the ADM trap server and the FTAC transfer admission remain unchanged.

DESTINATION = *NONE
The name of the ADM trap server and the FTAC transfer admission are deleted and thus reset to *NONE.

DESTINATION = *PARAMETERS(...)
Destination to which the ADM traps are to be sent.

PARTNER = *UNCHANGED
The name of the ADM trap server remains unchanged.

PARTNER = <text 1..200 with-low>
Name of the partner system from the partner list or the address of the partner system to which the ADM traps are to be sent. If the partner is not entered in the partner list, it must be specified with the prefix ftadm://. See section “Defining partner properties” on page 108.

TRANSFER-ADMISSION =
FTAC transfer admission for accessing the ADM trap server.

TRANSFER-ADMISSION = *UNCHANGED
The FTAC transfer admission of the ADM trap server remains unchanged.

**TRANSFER-ADMISSION = <alphanum-name 8..32> / <
c-string 8..32 with-low> / <x-string15..64>**
The FTAC functionality is used on the remote system. Only the transfer admission defined in the admission profile may be used.

SELECTION =
Activates or deactivates specific ADM traps. The default value is *NONE.

SELECTION = *UNCHANGED
The previous value is unchanged.

SELECTION = *NONE
Deactivates all ADM traps.

SELECTION = *ALL
Activates all ADM traps.

SELECTION = *PARAMETERS(...)
Activates or deactivates selected ADM traps.

FT-STATE =
Activates or deactivates the sending of traps on FTSTART / FTSTOP and if openFT is terminated abnormally.

FT-STATE = *UNCHANGED
The previous value is unchanged.

FT-STATE = *OFF
Deactivates the traps for FT-STATE.

FT-STATE = *ON
Activates the traps for FT-STATE.

PARTNER-STATE =
Activates or deactivates the sending of traps when the status of partners changes.

PARTNER-STATE = *UNCHANGED
The previous value is unchanged.

PARTNER-STATE = *OFF
Deactivates the traps for PARTNER-STATE.

PARTNER-STATE = *ON
Activates the traps for PARTNER-STATE.

PARTNER-UNREACHABLE =
Activates or deactivates the sending of the trap indicating that a partner is unreachable.

PARTNER-UNREACHABLE = *UNCHANGED
The previous value is unchanged.

PARTNER-UNREACHABLE = *OFF
Deactivates the "partner unreachable" trap.

PARTNER-UNREACHABLE = *ON
Activates the "partner unreachable" trap.

REQUEST-QUEUE-STATE =
Activates the sending of traps referring to the filling level of the request queue, i.e. whether traps are sent if the filling level has exceeded the 85% threshold or fallen below the 80% threshold.

REQUEST-QUEUE-STATE = *UNCHANGED
The previous value is unchanged.

REQUEST-QUEUE-STATE = *OFF
Deactivates the traps if the filling level falls outside the thresholds.

REQUEST-QUEUE-STATE = *ON
Activates the traps if the filling level falls outside the thresholds.

TRANSFER-SUCCESS =
Activates or deactivates the sending of the trap indicating that an FT request was completed successfully.

TRANSFER-SUCCESS = *UNCHANGED
The previous value is unchanged.

TRANSFER-SUCCESS = *OFF
Deactivates the trap for TRANSFER-SUCCESS.

TRANSFER-SUCCESS = *ON
Activates the trap for TRANSFER-SUCCESS.

TRANSFER-FAILURE =
Activates or deactivates the sending of the trap indicating that an FT request was aborted.

TRANSFER-FAILURE = *UNCHANGED
The previous value is unchanged.

TRANSFER-FAILURE = *OFF
Deactivates the trap for TRANSFER-FAILURE.

TRANSFER-FAILURE = *ON
Activates the trap for TRANSFER-FAILURE.

Example

The maximum number of tasks to be executed in parallel is to be 3 and the maximum number of transport connections to be set up is to be 10:

```
FTMODOPT PROCESS-LIMIT=3,CONNECTION-LIMIT=10
```

6.21 FTMODPRF Modify admission profile

User instruction

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

The command FTMODPRF can be used by any FTAC user to modify his admission profile. In a privileged admission profile, an FTAC user can only modify the operands TRANSFER-ADMISSION and PRIVILEGED.

Under certain circumstances, the FTAC administrator may modify external admission profiles:

- The FTAC administrator possesses the SU privilege (see page 62). He can then modify profiles for other user IDs without restriction.
- If the FTAC administrator does not possess the SU privilege but specifies ACCOUNT and PASSWORD in the USER-ADMISSION parameter, then he may also modify admission profiles. The TRANSFER-ADMISSION is only valid for as long as the current password for the user ID corresponds to the one defined in the profile.
- If the FTAC administrator does not possess the SU privilege and also does not specify the user’s password then he may only modify the PRIVILEGED property of external admission profiles.

The FTAC administrator can privilege foreign admission profiles with the operand PRIVILEGED.

You can also set up an admission profile for “pre-processing” or “post-processing”. When doing this, the operand FILE-NAME must begin with the pipe character ‘|’. Afterwards, one or several TSO commands may be specified. For detailed information refer to the section of the same name in the User Guide.

As soon as an admission profile is modified, the timestamp is also updated. The timestamp is output with FTSHWPRF INF=*ALL (LAST-MODIF). The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter FTMODPRF with the parameter NAME, but no other parameters.

Format

(part 1 of 2)

```
FTMODPRF

 NAME = *ALL / *STD / <alphanum-name 1..8>
,PASSWORD = *NONE / <alphanum-name 1..8>
,SELECT-PARAMETER = *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
        TRANSFER-ADMISSION = *ALL / *NOT-SPECIFIED / <alphanum-name 8..32> /
                                     <c-string 8..32 with-low> / <x-string 15..64>
       ,OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
,NEW-NAME = *OLD / *STD / <alphanum-name 1..8>
,TRANSFER-ADMISSION = *UNCHANGED / *NOT-SPECIFIED / *OLD-ADMISSION(...) /
                        <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
   *OLD-ADMISSION(...)
        VALID = *UNCHANGED / *YES / *NO
       ,USAGE = *UNCHANGED / *PRIVATE / *PUBLIC
       ,EXPIRATION-DATE = *UNCHANGED / *NOT-RESTRICTED / <date 8..10>
   <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
        VALID = *YES / *NO / *UNCHANGED
       ,USAGE = *PRIVATE / *PUBLIC / *UNCHANGED
       ,EXPIRATION-DATE = *NOT-RESTRICTED / <date 8..10> / *UNCHANGED
,PRIVILEGED = *UNCHANGED / *NO / *YES
,IGNORE-MAX-LEVELS = *UNCHANGED / *NO / *YES / *PARAMETERS(...)
   *PARAMETERS(...)
        OUTBOUND-SEND = *UNCHANGED / *NO / *YES
       ,OUTBOUND-RECEIVE = *UNCHANGED / *NO / *YES
       ,INBOUND-SEND = *UNCHANGED / *NO / *YES
       ,INBOUND-RECEIVE = *UNCHANGED / *NO / *YES
       ,INBOUND-PROCESSING = *UNCHANGED / *NO / *YES
       ,INBOUND-MANAGEMENT = *UNCHANGED / *NO / *YES
,USER-ADMISSION = *UNCHANGED / *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
        USER-IDENTIFICATION = *OWN / <name 1..8>
       ,ACCOUNT = *OWN / *NOT-SPECIFIED / <alphanum-name 1..40> / <c-string 1..40>
       ,PASSWORD = *OWN / *NOT-SPECIFIED / <alphanum-name 1..8> / *NONE
```

(part 2 of 2)

```
,INITIATOR = *UNCHANGED / list-poss(2): *REMOTE / *LOCAL

,TRANSFER-DIRECTION = *UNCHANGED / *NOT-RESTRICTED / *FROM-PARTNER / *TO-PARTNER

,PARTNER = *UNCHANGED / *NOT-RESTRICTED / *ADD(...) / *REMOVE(...) /
                    list-poss(50): <text 1..200 with-low>

   *ADD(...)
   |    NAME = list-poss(50): <text 1..200 with-low>

   *REMOVE(...)
   |    NAME = list-poss(50): <text 1..200 with-low>

,MAX-PARTNER-LEVEL = *UNCHANGED / *NOT-RESTRICTED / <integer 0..100>

,FILE-NAME = *UNCHANGED / *NOT-RESTRICTED  / <filename1..59> / <c-string 1..512 with-low> /

              *EXPANSION(...)

   *EXPANSION(...)
   |    PREFIX = <filename 1..58> / <filename-prefix 2..50> / <c-string 1..511 with-low>

,FILE-PASSWORD = *UNCHANGED / *NOT-RESTRICTED / *NONE / <alphanum-name 1..8>

,PROCESSING-ADMISSION = *UNCHANGED / *SAME / *NOT-RESTRICTED / *PARAMETERS(...)

   *PARAMETERS(...)
   |    USER-IDENTIFICATION = *SAME / *NOT-RESTRICTED / <name 1..8>
   |   ,ACCOUNT = *SAME / *NOT-RESTRICTED / <alphanum-name 1..40> / <c-string 1..40>
   |   ,PASSWORD = *SAME / *NOT-RESTRICTED / *NONE / <alphanum-name 1..8>

,SUCCESS-PROCESSING = *UNCHANGED / *NOT-RESTRICTED / *NONE / <c-string 1..1000 with-low> /
                        *EXPANSION(...)

   *EXPANSION(...)
   |    PREFIX = *UNCHANGED / *NOT-RESTRICTED / <c-string 1..999 with-low>
   |   ,SUFFIX = *UNCHANGED / *NOT-RESTRICTED / <c-string 1..999 with-low>

,FAILURE-PROCESSING = *UNCHANGED / *NOT-RESTRICTED / *NONE / <c-string 1..1000 with-low> /
                        *EXPANSION(...)

   *EXPANSION(...)
   |    PREFIX = *UNCHANGED / *NOT-RESTRICTED / <c-string 1..999 with-low>
   |   ,SUFFIX = *UNCHANGED / *NOT-RESTRICTED / <c-string 1..999 with-low>

,WRITE-MODE = *UNCHANGED / *NOT-RESTRICTED / *NEW-FILE / *REPLACE-FILE / *EXTEND-FILE

,FT-FUNCTION = *UNCHANGED / *NOT-RESTRICTED / list-poss(5):
                 *TRANSFER-FILE / *MODIFY-FILE-ATTRIBUTES / *READ-DIRECTORY /
                 *FILE-PROCESSING / *REMOTE-ADMINISTRATION

,USER-INFORMATION = *UNCHANGED / *NONE / <c-string 1..100 with-low>

,DATA-ENCRYPTION = *UNCHANGED / *NOT-RESTRICTED / *NO / *YES
```

Operands

NAME =
Determines the name of the admission profile to be modified.

NAME = *ALL
Modifies all your admission profiles at the same time.

NAME = *STD
Changes the default admission profile for your user ID.

NAME = <alphanum-name 1..8>
Modifies the admission profile with this name.

PASSWORD =
FTAC password which authorizes you to use FTAC commands on your user ID, if such a password has been defined in your admission set.

PASSWORD = <u>*NONE</u>
No FTAC password is required.

PASSWORD = <alphanum-name 1..8>
This FTAC password is required.

SELECT-PARAMETER =
Specifies a transfer admission. You will then modify the admission profile which has this transfer admission.

SELECT-PARAMETER = <u>*OWN</u>
Modifies your own admission profile.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies the selection criteria for the profiles which you wish to modify.

> **TRANSFER-ADMISSION =**
> Entering the TRANSFER-ADMISSION here makes it a selection criterion for the admission profiles which you wish to modify.
>
> **TRANSFER-ADMISSION = <u>*ALL</u>**
> All your admission profiles are to be modified, irrespective of the transfer admission.
>
> **TRANSFER-ADMISSION = *NOT-SPECIFIED**
> Only admission profiles without a defined transfer admission are to be modified. In the case of a default admission profile, the transfer admission is never assigned, because this is addressed using the user ID and the user password.
>
> **TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>**
> The admission profile with this transfer admission is to be modified.

OWNER-IDENTIFICATION =
You can use the owner of an admission profile as a selection criterion for access to a profile to be modified.

OWNER-IDENTIFICATION = *OWN
Modifies your own admission profile

OWNER-IDENTIFICATION = *ALL
The FTAC administrator can access the profiles of all users.
The FTAC user is not permitted to make this entry.

OWNER-IDENTIFICATION = <name 1..8>
The FTAC user can enter only his own user ID here, the FTAC administrator can enter any user ID.

NEW-NAME =
Gives your admission profile a new name (or not).

NEW-NAME = *OLD
The name of the admission profile remains unchanged.

NEW-NAME = *STD
Makes the admission profile the default admission profile for the user ID. If the admission profile previously had a transfer admission, you must also specify TRANSFER-ADMISSION=*NOT-SPECIFIED.

NEW-NAME = <alphanum-name 1..8>
New name of the admission profile. This name must be unique among all the admission profiles on your user ID. If an admission profile with this name already exists, FTAC rejects the command with the following message:

```
FTC0100   COMMAND REJECTED. FT-PROFILE ALREADY EXISTS
```

The command FTSHWPRF (see page 346ff) can be used to obtain information on the already existing name. For this information, it suffices to enter FTSHWPRF without parameters.

TRANSFER-ADMISSION =
Modifies the transfer admission which is associated with an admission profile. You must ensure that the transfer admission is unique within your openFT system. If the transfer admission which you have selected already exists, FTAC rejects the command with the following message:

```
FTC0101   COMMAND REJECTED. TRANSFER-ADMISSION ALREADY EXISTS
```

The FTAC administrator can also allocate an access admission here if he/she modifies the admissions profile of any user ID. To do this, however, the FTAC administrator must specify the complete USER-ADMISSION for the affected user ID (USER-IDENTIFICATION, ACCOUNT, and PASSWORD).

TRANSFER-ADMISSION = <u>*UNCHANGED</u>
The transfer admission remains unchanged.

TRANSFER-ADMISSION = *NOT-SPECIFIED
No transfer admission is set and any existing transfer admissions are made invalid. This blocks the profile, provided that it is not a profile that you are converting to a default admission profile. In this case, you must specify *NOT-SPECIFIED.

TRANSFER-ADMISSION = *OLD-ADMISSION(...)
The transfer admission itself remains unchanged. The options, however, can be changed, as opposed to with the entry TRANSFER-ADMISSION=*UNCHANGED. The specifications are ignored if you are changing a default admission profile.

VALID = <u>*UNCHANGED</u>
The value remains unchanged.

VALID = *YES
The transfer admission is valid.

VALID = *NO
The transfer admission is not valid. The profile can be blocked with this entry.

USAGE = <u>*UNCHANGED</u>
The value remains unchanged.

USAGE = *PRIVATE
Access to your profile is denied for security reasons whenever another user ID attempts to set for a second time the TRANSFER-ADMISSION which has already been used by you.

USAGE = *PUBLIC
Access to your profile is not denied if another user happens to “discover” your TRANSFER-ADMISSION. “Discovery” means that another user ID attempted to specify the same TRANSFER ADMISSION twice. This is rejected for security reasons.

EXPIRATION-DATE = <u>*UNCHANGED</u>
The value remains unchanged.

EXPIRATION-DATE = *NOT-RESTRICTED
The use of this transfer admission is not restricted with respect to time.

EXPIRATION-DATE = <date 8..10>
The use of the transfer admission is only possible until the given date (max. 19.01. 2038). The entry must be made in the form *yyyy-mm-dd* or *yy-mm-dd*.

TRANSFER-ADMISSION = <alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
The character string must be entered as transfer admission in the transfer request. The alphanumeric input is always stored in lowercase letters.

VALID = *YES
The transfer admission is valid.

VALID = *NO
The transfer admission is not valid. The profile can be blocked with this entry.

VALID = *UNCHANGED
The value remains unchanged.

USAGE = *PRIVATE
Access to your profile is denied for security reasons whenever another user ID attempts to set for a second time the TRANSFER-ADMISSION which has already been used by you.

USAGE = *PUBLIC
Access to your profile is not denied if another user happens to “discover” your TRANSFER-ADMISSION. “Discovery” means that another user ID attempted to specify the same TRANSFER ADMISSION twice. This is rejected for security reasons.

USAGE = *UNCHANGED
The value remains unchanged.
The use of this transfer admission is not restricted with respect to time.

EXPIRATION-DATE = *NOT-RESTRICTED
The use of this transfer admission is not restricted with respect to time.

EXPIRATION-DATE = <date 8..10>
The use of the transfer admission is only possible until the given date (max. 19.01.2038). The entry must be made in the form *yyyy-mm-dd* or *yy-mm-dd*.

EXPIRATION-DATE = *UNCHANGED
The value remains unchanged.

PRIVILEGED =
The FTAC administrator can privilege the admission profile of any FTAC user. FT requests which are processed with a privileged status are not subject to the restrictions for MAX-ADM-LEVEL in the admission set.
The FTAC user can only reverse any privileged status given.

PRIVILEGED = *UNCHANGED
The status of this admission profile remains unchanged.

PRIVILEGED = *NO
With *NO, you can reverse the privileged status.

IGNORE-MAX-LEVELS =
Determines for which of the six basic functions the restrictions of the admission set should be ignored. The user’s MAX-USER-LEVELS can be exceeded in this way. The MAX-ADM-LEVELS in the admission set can only be effectively exceeded with an admission profile which has been designated as privileged by the FTAC administrator. The FTAC user can set

up an admission profile for himself for special tasks (e.g. sending a certain file to a partner system with which he normally is not allowed to conduct a file transfer), which allows him to exceed the admission set. This profile must be explicitly given privileged status by the FTAC administrator.

If you enter IGNORE-MAX-LEVELS=*YES, the settings for all the basic functions are ignored. If you wish to ignore the admission set for specific basic functions, you need to do this with the operands explained later in the text.

The following table shows which partial components of the file management can be used under which conditions:

Inbound file management function	**Setting in admission set/extension in profile**
Show file attributes	Inbound sending (IBS) permitted
Modify file attributes	Inbound receiving (IBR) **and** Inbound file management (IBF) permitted
Rename files	Inbound receiving (IBR) **and** Inbound file management (IBF) permitted
Delete files	Inbound receiving (IBR) permitted **and** write rule = overwrite in profile
Show directories	Inbound file management (IBF) permitted **and** direction = to partner in profile
Create, rename, delete directories	Inbound file management (IBF) permitted **and** direction = from partner in profile

IGNORE-MAX-LEVELS = *UNCHANGED
You can access the same security levels as before the modification (unless you have reversed the privileged status with PRIVILEGED=*NO).

IGNORE-MAX-LEVELS = *NO
FT requests which are processed with the admission profile are subject to the restrictions of the admission set.

IGNORE-MAX-LEVELS = *YES
*YES allows you to communicate with partner systems whose security level exceeds the specifications of the admission set. If your profile does not have privileged status, you can only disregard the MAX-USER-LEVELS in the admission set, not the MAX-ADM-LEVELS. The current MAX-USER-LEVELS and MAX-ADM-LEVELS settings can be accessed using the command SHOW-FT-ADMISSION-SET (see example on page 306).

IGNORE-MAX-LEVELS = *PARAMETERS(...)

OUTBOUND-SEND = *UNCHANGED
The maximum security level which can be reached with the basic function “outbound send” remains unchanged.

OUTBOUND-SEND = *NO
The maximum security level which can be reached with the basic function “outbound send” is determined by the admission set.

OUTBOUND-SEND = *YES
For the basic function “outbound send”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

OUTBOUND-RECEIVE = <u>*UNCHANGED</u>
The maximum security level which can be reached with the basic function “outbound receive” remains unchanged.

OUTBOUND-RECEIVE = *NO
The maximum security level which can be reached with the basic function “outbound receive” is determined by the admission set.

OUTBOUND-RECEIVE = *YES
For the basic function “outbound receive”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-SEND = <u>*UNCHANGED</u>
The maximum security level which can be reached with the basic function “inbound send” remains unchanged.

INBOUND-SEND = *NO
The maximum security level which can be reached with the basic function “inbound send” is determined by the admission set.

INBOUND-SEND = *YES
For the basic function “inbound send”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The same applies to the partial component “display file attributes” of the basic function “inbound file management” can be used.

INBOUND-RECEIVE = <u>*UNCHANGED</u>
The maximum security level which can be reached with the basic function “inbound receive” remains unchanged.

INBOUND-RECEIVE = *NO
The maximum security level which can be reached with the basic function “inbound receive” is determined by the admission set.

INBOUND-RECEIVE = *YES
Disregards your settings for “inbound receive” in the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The same applies to the following partial components of the basic function “inbound file management”:

- delete files, as long as the file attributes are set accordingly,
- modify file attributes, if the basic function “inbound file management” was admitted in the admission set or in the admission profile.

INBOUND-PROCESSING = <u>*UNCHANGED</u>
The maximum security level which can be reached with the basic function “inbound processing” remains unchanged.

INBOUND-PROCESSING = *NO
The maximum security level which can be reached with the basic function “inbound processing” is determined by the admission set.

INBOUND-PROCESSING = *YES
For the basic function “inbound processing”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS.

INBOUND-MANAGEMENT = <u>*UNCHANGED</u>
The maximum security level which can be reached with the basic function “inbound file management” remains unchanged.

INBOUND-MANAGEMENT = *NO
The maximum security level which can be reached with the basic function “inbound file management” is determined by the admission set.

INBOUND-MANAGEMENT = *YES
For the basic function “inbound file management”, you can use this admission profile to disregard the MAX-USER-LEVELS. If your profile is privileged, you are also not held to the restrictions of the MAX-ADM-LEVELS. The partial component “modify file attributes” of the basic function “inbound file management” only functions if the basic function “inbound receive” was admitted in the admission set or admission profile.

USER-ADMISSION =
User ID under which the modified admission profile is saved. FT requests which use this profile access the entered user ID in the local system.
If the FTAC administrator has created an admission profile for a user, the user must enter the account and password in the operands ACCOUNT and PASSWORD described below before the profile can be used (see FTCREPRF, page 212).

USER-ADMISSION = <u>*UNCHANGED</u>
The USER-ADMISSION of this admission profile remains unchanged.

USER-ADMISSION = *OWN
For USER-IDENTIFICATION and ACCOUNT, the specifications for your user ID and your account number are taken from your LOGON authorization. A z/OS password is only taken from your LOGON authorization when an FT request accesses the admission profile.

Admission profiles in which USERID, ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for pre-processing, post-processing or follow-up processing. For pre-processing and post-processing, these parameters must be explicitly assigned a value in USER-ADMISSION. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

USER-ADMISSION = *PARAMETERS(...)
You can also enter the individual components of the user ID. This allows you to keep FT requests which use this admission profile under a different account number, for example. Or, a password can be set in the admission profile. FT requests which use this admission profile will then only function if their current LOGON password corresponds to the pre-set password.
The FTAC administrator can use these specifications to modify other users´ admission profiles.

USER-IDENTIFICATION =
Your user ID in z/OS

USER-IDENTIFICATION = <ins>*OWN</ins>
Your user ID is taken from your LOGON authorization.

USER-IDENTIFICATION = <name 1..8>
User ID with which the profile is to be associated.

ACCOUNT =
Account number under which an FT request is to be kept when it uses this admission profile.

ACCOUNT = <ins>*OWN</ins>
The account number is taken from your LOGON authorization.
Admission profiles in which ACCOUNT is set to its default value via *OWN cannot be used for pre-processing, post-processing or follow-up processing. For pre-processing and post-processing, this parameter must be explicitly assigned a value. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

ACCOUNT = *NOT-SPECIFIED
The account number is to be specified by the owner of the admission profile. This function permits the FTAC administrator to set up profiles for user IDs whose account numbers he doesn’t know.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
An FT request should be kept under the account number specified when it accesses this admission profile. You can enter any account number which is associated with your user ID.
You can also specify accounting information containing the account number which is to be used.

PASSWORD =
z/OS password associated with your user ID.

PASSWORD = *OWN
When an FT request refers to this admission profile, FTAC uses the BS2000 password valid for your user ID at that moment. This prevents you from having to modify the admission profile if the BS2000 password is changed.
Admission profiles in which PASSWORD is set to its default value via *OWN cannot be used for pre-processing, post-processing or follow-up processing. For pre-processing and post-processing, this parameter must be explicitly assigned a value. For follow-up processing, a specification in PROCESSING-ADMISSION is also possible.

PASSWORD = *NOT-SPECIFIED
The password is specified by the owner of the admission profile. This function permits the FTAC administrator to set up profiles for unknown user IDs.

PASSWORD = <alphanum-name 1..8>
When an FT request accesses the admission profile, the specified password is compared with the current LOGON password. If the two do not correspond, the FT request is rejected.

PASSWORD = *NONE
No password is required for the user ID.

INITIATOR =
Determines if initiators from local and/or remote systems are permitted to use this admission profile for their FT requests.

INITIATOR = *UNCHANGED
The settings in this admission profile remain unchanged,

INITIATOR = *REMOTE
This admission profile may only be used for FT requests by initiators from remote systems.

INITIATOR = *LOCAL
This admission profile may only be used for FT requests by initiators from the local system.

INITIATOR = (*LOCAL,*REMOTE)
This admission profile may be used by initiators from local and remote systems.

TRANSFER-DIRECTION =
Determines which transfer direction may be used with this admission profile.

i The transfer direction is always determined from the system in which the admission profile was defined.

TRANSFER-DIRECTION = *UNCHANGED
The specification in the admission profile remains unchanged.

TRANSFER-DIRECTION = *NOT-RESTRICTED
Files can be transferred to and from a partner system.

TRANSFER-DIRECTION = *FROM-PARTNER
Files can only be transferred from a partner system to your system. It is not possible to display file attributes/directories (partial components of “inbound file management”).

TRANSFER-DIRECTION = *TO-PARTNER
Files can only be transferred from your system to a partner system. It is not possible to modify file attributes or delete files (partial components of “inbound file management”).

PARTNER =
Specifies that this admission profile is to be used only for FT requests which are processed by a a certain partner system.

PARTNER = *UNCHANGED
Any partner in the admission profile remains unchanged.

PARTNER = *NOT-RESTRICTED
This admission profile’s scope of use is not limited to FT requests with certain partner systems.

PARTNER = *ADD(NAME = list-poss(50): <text 1..200 with-low>)
With this specification, you can add elements to an existing list of partner systems.
A maximum of 50 partner systems can be specified.

PARTNER = *REMOVE(NAME = list-poss(50): <text 1..200 with-low>)
Removes elements from an existing list of partner systems. A maximum of 50 partner systems can be specified.

PARTNER = list-poss(50): <text 1..200 with-low>
The admission profile only permits those FT requests which are processed with the specified partner systems. A maximum of 50 partner systems can be specified.
For PARTNER you can specify the name from the partner list or the address of the partner system, see also section “Specifying partner addresses” on page 108. You are advised to use the name from the partner list.

MAX-PARTNER-LEVEL =
A maximum security level can be specified. The admission profile will then only permit those FT requests which are processed with partner systems which have this security level or lower.

MAX-PARTNER-LEVEL works in conjunction with the admission set. When non-privileged admission profiles are used, the access check is executed on the basis of the smallest specified value.

MAX-PARTNER-LEVEL = *UNCHANGED
The specification for MAX-PARTNER-LEVEL in this admission set remains unchanged.

MAX-PARTNER-LEVEL = *NOT-RESTRICTED
If FT requests are processed with this admission profile, then the highest accessible security level is determined by the admission set.

MAX-PARTNER-LEVEL = <integer 0..100>
All partner systems which have this security level or lower can be communicated with.

i When you set MAX-PARTNER-LEVEL=0, you prevent access to the admission profile (for the time being). No FT request can then be processed with this admission profile.

FILE-NAME =
Determines which files or library members under your user ID may be accessed by FT requests that use this admission profile.

FILE-NAME = *UNCHANGED
The specifications for FILE-NAME in this admission profile remain unchanged.

FILE-NAME = *NOT-RESTRICTED
The admission profile permits unrestricted access to all files and library members of the user ID.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Only the specified file may be accessed.
However, openFT is also able to generate unique filenames automatically, thus providing an easy way of avoiding conflicts. This is done by specifying the string %UNIQUE at the end of the filename which is predefined here (see the section “File names” in the User Guide).
When follow-up processing is specified, this file can be referenced with %FILENAME, %FILN or %FILX, see the User Guide.
You can also directly specify file transfer with pre- and post-processing here by entering the pipe symbol ’|’ followed by a command.

FILE-NAME =*EXPANSION(PREFIX = <filename 1..58> / <filename-prefix 2..50> / <c-string 1..511 with-low>)
Restricts access to a number of files which all begin with the same prefix. If a filename is entered in an FT request which uses this admission profile, FTAC sets the prefix defined with EXPANSION in front of this filename. The FT request is then permitted to access the file *PrefixFilename*.

Example

- PREFIX=DONALD.; An FT request in which the FILE-NAME=DUCK is specified accesses the file DONALD.DUCK.
- PREFIX=TOOLS.CLIST/; an FT request in which FILE-NAME=MEMBER01 was specified, then accesses the file TOOLS.CLIST(MEMBER01).

Please note that the part of a filename which is specified in the file transfer command still has to be of the type <filename>.

If you want to perform file transfer with pre- or post-processing, you should indicate this by entering the pipe symbol ‘|’ at the start of the prefix. The created FTAC profile can then be used only for file transfer with pre- or post-processing since the file name that is generated also starts with a ‘|’. The variable %TEMPFILE can also be used in the filename prefix. You can find detailed information on preprocessing and postprocessing in the section of the same name in the User Guide.

The maximum length of the entire pre- or post-processing command is limited to the maximum length of the file name. If several commands are specified, then they must be separated by a semicolon (‘;’).

Example

```
FILE-NAME = *EXP(C‘|Command1;Command2;Command3; ...‘)
```

If you specify a name prefix that starts with a pipe character with *EXP(PREFIX=...), the preprocessing or postprocessing command of the FT request must not contain any semicolons. If the preprocessing or postprocessing command nevertheless contains semicolons, it must be enclosed in '...' (single quotes) .

Special cases

- In the case of admission profiles which are to be used exclusively for the ftexec command you must specify a filename or filename prefix that starts with the character string ‘|ftexecsv’ (see FTCREPRF, “Example 3” on page 230).
- Specify the file name prefix '|*ftmonitor' for admission profiles that are exclusively used for monitoring. A profile of this sort can then be used in the openFT Monitor or in an ft or ncopy command from a Windows or Unix system (see “Example 2” on page 230).

FILE-PASSWORD =
You can enter a password for files into the admission profile. The FTAC functionality then only permits access to files which are protected with this password and to unprotected files. When a FILE-PASSWORD is specified in an admission profile, the password may no longer be specified in an FT request which uses this admission profile. This allows you to permit access to certain files to users in remote systems, without having to disclose the file passwords.

FILE-PASSWORD = *UNCHANGED
The specifications for FILE-PASSWORD in this admission profile remain unchanged.

FILE-PASSWORD = *NOT-RESTRICTED
Permits access to all files. If a password is set for a file, then it must be specified in the transfer request.

FILE-PASSWORD = *NONE
Only permits access to files without file passwords.

FILE-PASSWORD = <alphanum-name 1..8>
Only permits access to files which are protected with the password specified and to unprotected files. The password which has already been specified in the profile may not be repeated in the transfer request. PASSWORD=*NONE would be entered in this case!

FILE-PASSWORD = *SECRET
The system prompts you to enter the password. However, this does not appear on the screen.

PROCESSING-ADMISSION =
You can enter a user ID in your z/OS system. Any follow-up processing of an FT request will be executed under this user ID. With PROCESSING-ADMISSION in the admission profile, you do not need to disclose your LOGON authorization to partner systems for follow-up processing.

i Admission profiles in which ACCOUNT and/or PASSWORD in USER-ADMISSION are set to their default values via *OWN cannot be used for follow-up processing. For follow-up processing, these parameters must be explicitly assigned a value either in USER-ADMISSION or in PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *UNCHANGED
The PROCESSING-ADMISSION in this admission profile remains unchanged.

PROCESSING-ADMISSION = *SAME
For the PROCESSING-ADMISSION, the values of the USER-ADMISSION are used. If *SAME is entered here, then any FT request which uses this profile must also contain PROCESSING-ADMISSION=*SAME or PROCESSING-ADMISSION= *NOT-SPECIFIED.

PROCESSING-ADMISSION = *NOT-RESTRICTED
FT requests which use this admission profile may contain any PROCESSING-ADMISSION.

PROCESSING-ADMISSION = *PARAMETERS(...)
You can also enter the individual components of the user ID. This allows follow-up processing using this admission profile and started from FT requests to be charged under a different account number, for example. Or, a password can be set in the admission profile. Follow-up processing for FT requests which use this admission profile will then only function if their current LOGON password corresponds to the pre-set password.

USER-IDENTIFICATION =
User ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = *SAME
The USER-IDENTIFICATION is taken from the USER-ADMISSION.

USER-IDENTIFICATION = *NOT-RESTRICTED
The admission profile does not restrict the user ID under which the follow-up processing is to be executed.

USER-IDENTIFICATION = <name 1..8>
FT requests which are processed with this admission profile are only permitted follow-up processing under this user ID. If another user ID is entered here, the parameter PASSWORD must also be entered. PASSWORD=*SAME is then not valid.

ACCOUNT =
Specifies the account number for the follow-up processing.

ACCOUNT = *SAME
The account number is taken from the USER-ADMISSION.

ACCOUNT = *NOT-RESTRICTED
The account number may be specified in FT requests that work with the admission profile. The admission profile does not restrict the account for follow-up processing.

ACCOUNT = <alphanum-name 1..40> / <c-string 1..40>
Follow-up processing is to be settled under this account number.
You can also specify account information containing the account number to be used.

PASSWORD =
Specifies, where applicable, the z/OS password for the user ID under which the follow-up processing is to be executed. Here, you can enter a PASSWORD when the user ID in question doesn’t have such a password (yet).

PASSWORD = *SAME
The value *SAME is only valid if the PROCESSING-ADMISSION refers to your own user ID. If PASSWORD=*OWN is entered on USER-ADMISSION, then the BS2000 password valid at the time of the request is used for the PROCESSING-ADMISSION.

PASSWORD = *NOT-RESTRICTED
The password may be specified for FT requests which work with the admission profile. The admission profile does not restrict the password for follow-up processing.

PASSWORD = *NONE
FT requests which use this admission profile can only initiate follow-up processing on user IDs without a password.

PASSWORD = <alphanum-name 1..8>
FT requests which use the admission profile may only initiate follow-up processing on user IDs which are protected with this password.

SUCCESS-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a successful data transfer.

SUCCESS-PROCESSING = *UNCHANGED
The specifications for SUCCESS-PROCESSING in this admission profile remain unchanged.

SUCCESS-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand SUCCESS-PROCESSING may be used without restriction.

SUCCESS-PROCESSING = *NONE
The admission profile does not permit follow-up processing after successful data transfer.

SUCCESS-PROCESSING = <c-string 1..1000 with-low>
BS2000 commands which are executed in the local system after successful data transfer. The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

SUCCESS-PROCESSING = *EXPANSION(...)
If a SUCCESS-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = *UNCHANGED
The specifications for the follow-up processing prefix in this admission profile remain unchanged.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *UNCHANGED
The specifications for the follow-up processing suffix in this admission profile remain unchanged.

SUFFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified prefix is set after a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

Example

If PREFIX='SEND ' and SUFFIX=',USER(USER1)' is specified and SUCC="'FILE TRANSFER OK'" is defined in the FT request, FT executes the command "SEND 'FILE TRANSFER OK',USER(USER1)" for follow-up processing.

FAILURE-PROCESSING =
Restricts the follow-up processing which an FT request is permitted to initiate in your system after a failed data transfer.

FAILURE-PROCESSING = <u>*UNCHANGED</u>
The specifications for FAILURE-PROCESSING in this admission profile remain unchanged.

FAILURE-PROCESSING = *NOT-RESTRICTED
In FT requests which use this admission profile the operand FAILURE-PROCESSING may be used without restriction.

FAILURE-PROCESSING = *NONE
The admission profile does not permit follow-up processing after failed data transfer.

FAILURE-PROCESSING = <c-string 1..1000 with-low>
z/OS commands which are executed in the local system after failed data transfer. Individual commands must be preceded by a slash (/). The individual commands must be separated by a semicolon (;). If a character string is enclosed by single or double quotes (’ or ”) within a command sequence, openFT does not interpret any semicolons within this character string as a separator.

FAILURE-PROCESSING = *EXPANSION(...)
If a FAILURE-PROCESSING was specified in an FT request which uses this admission profile, FTAC adds the prefix or suffix specified here to this command. As follow-up processing, the command which has been thus expanded is then executed.

If a suffix or prefix is defined at this point, then no command sequence for the follow-up processing may be specified in FT requests which use this admission profile. This makes the setting of prefixes and suffixes mandatory.

PREFIX = <u>*UNCHANGED</u>
The specifications for the follow-up processing prefix in this admission profile remain unchanged.

PREFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a prefix.

PREFIX = <c-string 1..999 with-low>
The specified prefix is set in front of a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the prefix is executed as follow-up processing.

SUFFIX = *UNCHANGED
The specifications for the follow-up processing suffix in this admission profile remain unchanged.

SUFFIX = *NOT-RESTRICTED
Follow-up processing is not restricted by a suffix.

SUFFIX = <c-string 1..999 with-low>
The specified prefix is set after a command which is specified in an FT request as follow-up processing. Then, the command which has been expanded with the suffix is executed as follow-up processing.

WRITE-MODE =
Determines the WRITE-MODE which is valid for this FT request. WRITE MODE is only effective if the receive file is in the same system as the admission profile definition.

WRITE-MODE = *UNCHANGED
The specifications for WRITE-MODE in this admission profile remain unchanged.

WRITE-MODE = *NOT-RESTRICTED
In an FT request which accesses this admission profile, WRITE-MODE may be used without restrictions.

WRITE-MODE = *NEW-FILE
In the FT request, *NEW-FILE, *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. If the receive file already exists, the transfer will be rejected.

WRITE-MODE = *REPLACE-FILE
In the FT request of openFT partners, only *REPLACE-FILE or *EXTEND-FILE may be entered for WRITE-MODE. With ftp partners, *NEW-FILE may also be entered if the file does not yet exist.

WRITE-MODE = *EXTEND-FILE
In the FT request, only *EXTEND-FILE may be entered for WRITE-MODE.

FT-FUNCTION =
This operand permits the restriction of the profile validity to certain FT functions (=file transfer and file management functions).

FT-FUNCTION = *UNCHANGED
The previous scope of the FT functions remains unchanged.

FT-FUNCTION = *NOT-RESTRICTED
The full scope of FT functions is available with the exception of the “remote administration” function (*REMOTE-ADMINISTRATION). This must be activated explicitly.

FT-FUNCTION = (*TRANSFER-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY, *FILE-PROCESSING, *REMOTE-ADMINISTRATION)
The following file transfer functions are available:

***TRANSFER-FILE**
The admission profile may be used for the file transfer functions “transfer files”, “view file attributes” and “delete files”.

***MODIFY-FILE-ATTRIBUTES**
The admission profile may be used for the file transfer functions “view file attributes” and “modify file attributes”.

***READ-DIRECTORY**
The admission profile may be used for the file transfer functions “view directories” and “view file attributes”.

***FILE-PROCESSING**
The admission profile may be used for the “pre-processing” and “post-processing” file transfer functions. The “transfer files” function must also be permitted.

The *FILE-PROCESSING specification is of relevance only for FTAC profiles without a filename prefix. Otherwise the first character of the filename prefix determines whether only normal data transfer (no pipe symbol “|”) or only pre- and post-processing (pipe symbol “|”) are to be possible with this FTAC profile.

***REMOTE-ADMINISTRATION**
The admission profile is allowed to be used for the "remote administration" function. This allows a remote administrator to administer the openFT instance using this profile. *REMOTE-ADMINISTRATION may only be specified by the FT administrator or FTAC administrator.

USER-INFORMATION =
Specifies a text in the admission profile. This text can be displayed with the FTSHWPRF command.

USER-INFORMATION = <u>*UNCHANGED</u>
Any existing text remains unchanged.

USER-INFORMATION = *NONE
Any existing text is deleted.

USER-INFORMATION = <c-string 1..100 with-low>
The character string entered is accepted as user information.

DATA-ENCRYPTION =
Specifies whether user data with this profile must be transferred in encrypted form.

DATA-ENCRYPTION = <u>*UNCHANGED</u>
The encryption option should remain unchanged.

DATA-ENCRYPTION = *NOT-RESTRICTED
The encryption option for user data is not restricted. File transfer requests with encryption and file transfer requests without encryption are both accepted

DATA-ENCRYPTION = *NO
Only file transfer requests that do not have encrypted user data are accepted, i.e. requests with encryption are rejected.
If the request is made in a BS2000 or z/OS, DATA-ENCRYPTION=*NO must be specified there in the NCOPY request.

DATA-ENCRYPTION = *YES
Only file transfer requests that have encrypted user data are accepted, i.e. requests without encryption are rejected.
If the request is made in a BS2000 or z/OS, for example, then DATA-ENCRYPTION=*YES must be specified there in the NCOPY request.

When using restrictions for FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING, keep in mind that

- a restriction for follow-up processing must always be made for SUCCESS- and FAILURE-PROCESSING. Otherwise, it is possible that users will avoid this step.
- PREFIX of FILE-NAME, SUCCESS-PROCESSING and FAILURE-PROCESSING must correspond,
  e.g. FILE-NAME = *EXP(XYZ.),SUCC = *EXP(’PR DSNAME(XYZ.’,')')

Example

After Donald Duck has created an admission profile with the name *profile1*, which permits other users access to his user ID with the LOGON authorization, he decides he wants to restrict this profile so that only FT accesses are possible to files which begin with the prefix *BRANCH*.

The required command is:

```
FTMODPRF␣NAME=PROFIL1,
         FILE-NAME=*EXPANSION(PREFIX=BRANCH.)
```

A possible short form of this command is:

```
FTMODPRF␣PROFIL1,FILE-N=(PRE=BRANCH.)
```

This places heavy restrictions on the admission profile. The other specifications remain unchanged.

6.22 FTMODPTN
Modify partner properties in the partner list

Note on usage

User group: FT administrator

Functional description

This command can be used to modify the characteristics of a partner that is already entered in the partner list. **Format**

```
FTMODPTN
---------------------------------------------------------------------------
 PARTNER = *ALL / <text 1..200 with-low>
,STATE = *UNCHANGED / *ACTIVE(...) / *DEACT
   *ACTIVE(...)
     |  AUTOMATIC-DEACT = *NO / *YES
,SECURITY-LEVEL = *UNCHANGED / *STD / *BY-PARTNER-ATTRIBUTES / <integer 1..100>
,PARTNER-ADDRESS = *UNCHANGED / <text 1..200 with-low>
,TRACE = *UNCHANGED / *BY-FT-OPTIONS / *ON / *OFF
,IDENTIFICATION = *UNCHANGED / *STD / <composed-name 1..64> / <c-string 1..64 with-low>
,SESSION-ROUTING-INFO = *UNCHANGED / *NONE / *IDENTIFICATION / <alphanum-name 1..8>
,PARTNER-CHECK = *UNCHANGED / *BY-FT-OPTIONS / *STD / *TRANSPORT-ADDRESS
,AUTH-MANDATORY = *UNCHANGED / *NO / *YES
,PRIORITY= *UNCHANGED / *NORMAL / *LOW / *HIGH
```

Operands

PARTNER =
Specifies the partner system or systems.

PARTNER = *ALL
The specified changes are to be implemented for all partner systems defined in the partner list. This specification is only meaningful in conjunction with the operands STATE, SECURITY-LEVEL and TRACE. If the value of SECURITY-LEVEL is modified and if, simultaneously, dynamic partners are permitted, the value of the security level for dynamic partners remains unchanged and as set in the openFT options.

PARTNER = <text 1..200 with-low>
Specifies either the name of the partner system from the partner list or the address of the partner system (see section "Defining partner properties" on page 108).

STATE =
Controls FT requests issued locally which are directed to a particular remote system.

STATE = <u>*UNCHANGED</u>
The state is unchanged.

STATE = *ACTIVE(...)
Deactivated FT requests to a remote system, which were issued locally are released again for processing.

> **AUTOMATIC-DEACT =**
> Defines if repeated attempts to establish a connection with a partner system should result in a deactivation of the partner system after a certain number of consecutive unsuccessful attempts.
>
> **AUTOMATIC-DEACT = <u>*NO</u>**
> Unsuccessful attempts to establish a connection with this partner do not lead to its deactivation.
>
> **AUTOMATIC-DEACT = *YES**
> Repeated unsuccessful attempts to establish a connection with this partner lead to its deactivation. If you wish to run file transfers with this partner afterwards, it must be explicitly reactivated.

STATE = *DEACT
FT requests to a remote system, which were issued locally are accepted, but will only be executed when the remote system is reactivated.

SECURITY-LEVEL =
Assigns a security level to a remote system.

SECURITY-LEVEL = <u>*UNCHANGED</u>
The value is unchanged.

SECURITY-LEVEL = *STD
If you set this operand to *STD, a standard security level is assigned to the remote system. This standard security level is defined using the FTMODOPT command.

SECURITY-LEVEL = *BY-PARTNER-ATTRIBUTES
If you set the operand to *BY-PARTNER-ATTRIBUTES then the security level is defined automatically. This setting assigns partners that are authenticated by openFT the security level 10. Partners, known to the transport system, are assigned the security level 90. All other partners are assigned security level 100.

SECURITY-LEVEL = <integer 1..100>
Must be specified if you want to use FTAC functionality and assign a security level to an individual remote system

PARTNER-ADDRESS =
Address of the remote system.

PARTNER-ADDRESS = *UNCHANGED
The address remains unchanged.

PARTNER-ADDRESS = <text 1..200 with-low>
New address for the partner system. For details on the address format, see section "Defining partner properties" on page 108.

TRACE =
Trace setting for openFT partner systems. Trace entries are not generated unless TRACE=*ON is set in FTMODOPT.

TRACE = *UNCHANGED
The current trace setting is unchanged.

TRACE = *BY-FT-OPTIONS
The trace settings specified in the MODIFY-FT-OPTIONS command are used.

TRACE = *ON
Activates the trace for this partner even if tracing is deactivated for this partner type in the global settings (FTMODOPT). The request-specific trace settings made in FTMODOPT, on the other hand, are taken into account.

TRACE = *OFF
For connections to this partner system, only those trace entries which it is technically impossible to suppress are generated. Trace entries which it is technically impossible to suppress are those which are generated before openFT (BS2000) identifies the partner system

IDENTIFICATION =
The network-wide, unique ID of the openFT instance in the partner system.

IDENTIFICATION = *UNCHANGED
The ID remains unchanged.

IDENTIFICATION = *STD
For openFT and FTADM partners, the partner address or the host name from the partner address is used as the identification.

IDENTIFICATION = <composed-name 1..64> / <c-string 1..64 with-low>
The network-wide, unique instance ID of the openFT instance in the partner system. It is set by the FT administrator of the partner system (for example in BS2000, by using MODIFY-FT-OPTIONS IDENTIFICATION=, in Unix systems or Windows, by using ftmodo -

id). The uniqueness of this ID must be based on something other than case-sensitivity. An instance ID may be comprised of alphanumeric characters or special characters. It is advisable to use only the special characters “.”, “-”, “:” or “%”.

The initial character must be alphanumeric or the special character “%”. The “%” character may only be used as an initial character. An alphanumeric character must follow the “.” character. For more details on assigning instance identifications, see page 114.

SESSION-ROUTING-INFO =
If the partner system is addressed via IDENTIFICATION, but is only accessible via a go-between instance (e.g. an openFTIF gateway), specify here the address information, which the go-between instance will use for re-routing.

SESSION-ROUTING-INFO = *UNCHANGED
The setting remains unchanged.

SESSION-ROUTING-INFO = *NONE
No routing information is used.

SESSION-ROUTING-INFO = *IDENTIFICATION
Connections to the partner are re-routed via a gateway that supports IDENTIFICATION as the address information.

SESSION-ROUTING-INFO = <alphanum-name 1..8>
Connections to the partner are re-routed via a gateway, that uses the specified string as addressing information.

PARTNER-CHECK =
You can set (for the partner) an allocation for sender checking that deviates from the global settings.
PARTNER-CHECK is only effective for openFT partners that are not authenticated in the local system (see section “Authentication” on page 113).

PARTNER-CHECK = *UNCHANGED
The set value remains unchanged.

PARTNER-CHECK = *BY-FT-OPTIONS
The global settings are valid for the partner.

PARTNER-CHECK = *STD
Disable the expanded sender checking. The transport address of the partner is not checked, even if the expanded sender checking is globally enabled (see the FTMODOPT command).

PARTNER-CHECK = *TRANSPORT-ADDRESS
Enables expanded sender checking. The transport address is checked, even if the expanded sender checking is globally disabled (see the FTMODOPT command).
If the transport address under which the partner is reporting does not correspond to the entry in the partner list, the request is rejected.

AUTH-MANDATORY =
Forces the authentication of a partner.

AUTH-MANDATORY = *UNCHANGED
The set value is unchanged.

AUTH-MANDATORY = *NO
Authentication is not forced, i.e. this partner is not restricted with regard to authentication.

AUTH-MANDATORY = *YES
Authentication is forced, i.e. this partner may only work with authentication.

PRIORITY=
This operand allows you to specify the priority of a partner in respect of processing requests that have the same request priority. This means that the partner priority only applies in the case of requests that have the same request priority, but that are issued to partners with a different partner priority.

PRIORITY = *NORMAL
The partner has normal priority.

PRIORITY = *LOW
The partner has low priority.

PRIORITY = *HIGH
The partner has high priority.

Example 1

The SECURITY-LEVEL for the partner system TEST is set to 99:

```
FTMODPTN PARTNER=TEST,SECURITY-LEVEL=99
```

Example 2

The port number for partner WINDOWS (host name = winhost2) is set to 1100:

```
FTMODPTN WINDOWS,PARTNER-ADDRESS=winhost2:1100
```

6.23 FTMODREQ Modify request queue

Note on usage

User group: FT user and FT administrator

Functional description

You use the FTMODREQ command to modify the position and priority of your outbound requests within the openFT request queue. You have the option of processing the outbound requests in any order you wish. Newly input requests or requests whose priority changes are put at the end of the request queue for the corresponding priority. If already active requests are repositioned behind waiting outbound requests, the active requests are interrupted if possible in favor of those waiting.

FTMODREQ is only valid for outbound requests.

The sequence of requests with a starting time in the future cannot be modified.

Format

```
FTMODREQ

 TRANSFER-ID = *ALL / <integer 1..2147483639>
,SELECT = *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
     |  OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
     | ,PARTNER = *ALL / <text 1..200 with-low>
     | ,FILE = *ALL / <filename 1..59> / <c-string 1..512 with-low>
,QUEUE-POSITION = *UNCHANGED / *FIRST / *LAST
,PRIORITY = *UNCHANGED / *NORMAL / *HIGH / *LOW
```

Operands

TRANSFER-ID =
Transfer ID of the outbound request to be modified.

TRANSFER-ID = *ALL
Modifies all outbound requests, If further selections haven’t been specified with SELECT (see below).

TRANSFER-ID = <integer 1..2147483639>
Transfer ID which is communicated to the local system in the FT request confirmation.

SELECT =
Contains selection criteria for outbound requests to be modified. A request is only modified if all the criteria specified are met. This means that the criteria must be specified accordingly in the FT request.

SELECT = *OWN
Modifies all FT requests of the user's own ID.

SELECT = *PARAMETERS(...)

OWNER-IDENTIFICATION =
Identifies the owner of the FT request.

OWNER-IDENTIFICATION = *OWN
Modifies only outbound requests with the user's own ID.

OWNER-IDENTIFICATION = *ALL
Modifies outbound requests for all user IDs.
Only the FTAC administrator may use this entry.

OWNER-IDENTIFICATION = <name 1..8>
Specifies a user ID whose requests are to be modified.
Users may only enter their own user ID.

PARTNER =
Modifies outbound requests which are to be executed with a particular partner system.

PARTNER = *ALL
The name of the partner system is not selected as a criterion for the outbound requests to be modified.

PARTNER = <text 1..200 with-low>
Modifies outbound requests which are to be executed with this partner system. You can specify the name from the partner list or the address of the partner system. For more information on address specifications, see section "Specifying partner addresses" on page 108.

FILE =
Modifies outbound requests which access this file or library member in the local system as a send or receive file. The file or library member name must be entered exactly as in the file transfer request.

FILE = *ALL
The filename is not selected as a criterion for the outbound requests to be modified.

FILE = <filename 1..59> / <c-string 1..512 with-low>
Modifies outbound requests which access this file (DVS/POSIX) in the local system.

QUEUE-POSITION =
New position of the outbound request that is to be modified in the openFT request queue.

QUEUE-POSITION = *UNCHANGED
The position of the outbound request in this user's openFT request queue remains unchanged.

QUEUE-POSITION = *FIRST
The outbound request is placed in front of all the other requests of the same priority issued by the user in the openFT request queue.

QUEUE-POSITION = *LAST
The outbound request is placed behind all the other requests of the same priority issued by the user in the openFT request queue.

PRIORITY =
Modifies the priority of the FT request.

PRIORITY = *UNCHANGED
The priority of the FT request remains unchanged.

PRIORITY = *NORMAL
The priority of the FT request is set to the normal value

PRIORITY = *HIGH
The FT request is given a high priority.

PRIORITY = *LOW
The FT request is given a low priority.

Example

```
NSTATUS
  TRANS-ID  INI  STATE  PARTNER  DIR   BYTE-COUNT FILE-NAME
  54483612  LOC   WAIT  UNIX1    FROM  0          FILE1
  11164324  LOC   WAIT  UNIX2    FROM  0          FILE2
```

```
FTMODREQ SELECT=(FILE=FILE2),QUEUE-POS=*FIRST
```

```
NSTATUS
  TRANS-ID  INI  STATE  PARTNER  DIR   BYTE-COUNT FILE-NAME
  11164324  LOC   WAIT  UNIX2    FROM  0          FILE2
  54483612  LOC   WAIT  UNIX1    FROM  0          FILE1
```

6.24 FTREMPTN Remove remote system from partner list

Note on usage

User group: FT administrator

Functional description

The FTREMPTN command is used to remove a remote system from the partner list of the current openFT instance.

If a partner system is deleted from the partner list then all requests involving this partner system are aborted. FTREMPTN therefore represents a simple way to delete all the requests relating to a given partner. A request to a partner removed with FTREMPTN is eliminated even if the request is already known in the partner system (in the same way as with NCANCEL .. FORCE-CANCELLATION=*YES).

Format

FTREMPTN
PARTNER = <text 1..200 with-low>

Operands

PARTNER = <text 1..200 with-low>
Name of the partner system from the partner list or the address of the partner system. For details on specifying partner addresses, see page 108.

Example

Remove the remote system PARTNER1 from the partner list of the current openFT instance:

```
ftremptn partner1
```

6.25 FTSHWADS Display admission sets

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

You use the FTSHWADS command to display admission sets. You can output the following information on either SYSTSPRT or SYSPRINT:

- if the admission set is privileged (if so, then you are the FTAC administrator).
- if a password is required to use FTAC commands on this user ID. The password itself is not displayed.
- the limiting values for accessible security levels which have been set by the owner of this user ID.
- the limiting values for accessible security levels which have been pre-set by the FTAC administrator.

Format

FTSWHADS
USER-IDENTIFICATION = ***OWN** / ***ALL** / ***STD** / <name 1..8> ,**SEL**ECT-**PAR**AMETER = ***ALL** ,**OUTPUT** = ***STDERR**(...) / ***STDOUT**(...) ***STDERR**(...) / ***STDOUT**(...) │ **LAYOUT** = ***STD** / ***CSV**

Operands

USER-IDENTIFICATION =
User ID whose admission set you wish to view. FTAC users can only obtain information about their own admission set and the default admission set. The FTAC administrator can obtain information about any admission set.

USER-IDENTIFICATION = *OWN
FTAC outputs your own user ID’s admission set.

USER-IDENTIFICATION = *ALL
FTAC outputs the default admission set and the admission set of your own user ID.
For the FTAC administrator, all admission sets are output which differ from the default admission set.

USER-IDENTIFICATION = *STD
FTAC only outputs the default admission set.

USER-IDENTIFICATION = <name 1..8>
FTAC outputs the admission set of the user ID indicated. The operand stands for the USER-ID of the specified user ID. The FTAC user can only enter his own user ID here. The FTAC administrator can enter any user ID.

SELECT-PARAMETER = *ALL
SELECT-PARAMETER = *ALL
This parameter is reserved for future extensions and has no effect in the current version.

OUTPUT =
Output medium for the information requested.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stderr.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stdout.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**omma **S**eparated **V**alue) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see section “Output in CSV format” on page 181).

Example

Dagobert Duck, the FTAC administrator of the Duck Bank, wants to obtain information about the admission sets in his system. He enters the command

```
FTSHWADS␣USER-IDENTIFICATION=*ALL
```

Short form:

```
FTSHWADS␣*ALL
```

He receives the following output:

```
              MAX. USER LEVELS              MAX. ADM LEVELS          ATTR
USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
*STD      10   10   10   10    0    0   10   10   10   10    0    0
DAGO     100  100    0    0    0*   0* 100  100    0    0    0*   0* PRIV
DAISY     50   50   10*  50   50   50   50   50   50   50   50   50  PW
DANIEL     0   10    0    0    0    0   10   10    0    0    0    0  PW
DONALD    50  100    0   10*   0    0   50  100   10   50    0    0
```

These can be explained as follows:

The user ID of each admission set is in the column USER-ID. In this example, there is a default admission set as well as admission sets for the user IDs DAGO, DAISY, DANIEL and DONALD.

The column ATTR indicates the privileged admission set. We can see that DAGO is the FTAC administrator.

The column ATTR also indicates whether an FTAC password has been defined (with PW). DAGOBERT, DAISY and DANIEL have done this to prevent others from using FTAC commands on their user ID which could be used to make modifications.

In the six columns under MAX-USER-LEVELS, the limiting values are output which the FTAC users have set for their admission sets. The six columns under MAX-ADM-LEVELS show the limiting values which the FTAC administrator has set. The smaller of the two values indicates up to which security level the owner of the admission set may use each basic function. The basic functions are abbreviated in the output as follows:

OBS	=	**O**UT**B**OUND-**S**END
OBR	=	**O**UT**B**OUND-**R**ECEIVE
IBS	=	**I**N**B**OUND-**S**END
IBR	=	**I**N**B**OUND-**R**ECEIVE
IBP	=	**I**N**B**OUND-**P**ROCESSING
IBF	=	**I**N**B**OUND-**F**ILEMANAGEMENT

The default admission set is configured such that it permits file transfers with systems which have the security level of 10 or lower, but does not permit any follow-up processing initiated by external sources (IBP=0). DAGO may contact all available partner systems (OBS=100,OBR=100), but does not permit any file transfer accesses from outside onto his user ID (IBS=0,IBR=0,IBP=0).

The user ID DAISY is permitted to communicate with all partner systems with the security level of 50, according to the FTAC administrator's specifications. To better protect her files from strangers, DAISY has only made the function “inbound send” available to partner systems with the security level f 10 or lower.

The user ID DANIEL is heavily protected. Only files from partner systems with a maximum security level of 10 may be requested. A * after a number indicates that this value was taken from the default admission set and will change if any modifications are made to the default admission set.

6.26 FTSHWENV Display saved admission profiles and sets

Note on usage

User group: FTAC administrator

openFT-AC must be installed to use this command.

Functional description

The FTAC administrator can use the command FTSHWENV to view admission profiles and sets which have been written in an export file using the command FTEXPENV (see page 241). This function is particularly useful before the importing of the admission profiles and sets (see page 245).

This command may not be used by the FTAC user.

Format

FTSHWENV
__FROM-FILE__ = <filename 1..46>
__,USER-IDENTIFICATION__ = __*ALL__ / list-poss(100): <name 1..8>
__,SELECT-PARAMETER__ = __*ALL__ / __*PAR__AMETERS(...)
__*PAR__AMETERS(...)
__PROFILE-NAME__ = __*ALL__ / __*NONE__ / __*STD__ / list-poss(100): <alphanum-name 1..8>
__,ADMISSION-SET__ = __*YES__ / __*NO__
__,INFORMATION__ = __*ONLY__-NAMES / __*ALL__
__,OUTPUT__ = __*STDERR__(...) / __*STDOUT__(...)
__*STDERR__(...) / __*STDOUT__(...)
__LAYOUT__ = __*STD__ / __*CSV__

Operands

FROM-FILE = <filename 1..46>
Name of the file (not a temporary file) from which the admission profiles and sets are to be displayed. If the file contains invalid data or access to the file is unsuccessful, the command is rejected with the message FTC0103.

USER-IDENTIFICATION =
User ID whose admission profiles and sets are to be displayed.

USER-IDENTIFICATION = *ALL
The admission profiles and sets of all users are to be displayed.

USER-IDENTIFICATION = list-poss(100): <name 1..8>
The admission profiles and sets of the user IDs specified (maximum 100) are to be displayed.

SELECT-PARAMETER =
Specifies whether only admission profiles, only admission sets or both are to be displayed. For the admission profiles, you can specify which ones are to be displayed.

SELECT-PARAMETER = *ALL
Specifies whether only admission profiles, only admission sets or both are to be displayed. For the admission profiles, you can specify which ones are to be displayed.

SELECT-PARAMETER = *PARAMETERS(...)
Specifies which of the admission sets associated with the USER-IDENTIFICATION are to be specified.

PROFILE-NAME = *ALL
All admission profiles are displayed.

PROFILE-NAME = *NONE
No admission profiles are displayed.

PROFILE-NAME = *STD
Displays the default admission profile.

PROFILE-NAME = list-poss(100): <alphanum-name 1..8>
Only the specified profiles are displayed (maximum 100).

ADMISSION-SET = *YES
All admission sets are displayed.

ADMISSION-SET = *NO
No admission sets are displayed.

INFORMATION =
Scope of the information to be displayed.

INFORMATION = *ONLY-NAMES
Only the names of the admission profiles are to be displayed.

INFORMATION = *ALL
The entire contents of the admission profiles, excluding any passwords and transfer admissions, are displayed.

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stderr.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT. If the command is called with ftexec from a Unix or Windows system, ftexec sends the output to stdout.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user

LAYOUT = *CSV
Output is supplied in CSV (**C**omma **S**eparated **V**alue) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 181).

Example

The FTAC administrator Dagobert Duck backs up the admission set and the admission profiles of the user ID DONALD in the file DONALD.FTAC.BKUP.

```
FTEXPENV␣TO-FILE=DONALD.FTAC.BKUP,USER-IDENTIFICATION=DONALD
```

A possible short form of this command would be:

```
FTEXPENV␣DONALD.FTAC.BKUP,DONALD
```

As a conscientious FTAC administrator, Dagobert checks if the desired back-up is in the file DONALD.FTAC.BKUP

```
FTSHWENV␣FROM-FILE=DONALD.FTAC.BKUP
```

He receives the following output:

```
               MAX. USER LEVELS                  MAX. ADM LEVELS         ATTR
 USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
 DONALD     1    1    0    1    0    0    1    1    0    0    0    0
 OWNER     NAME
 DONALD   *UMSAWARE
```

USER-ID and OWNER can be used to determine the user ID with which the admission sets and profiles defined under NAME are associated.

In addition, the maximum security levels set for each user are displayed, as in the command FTSHWADS. An explanation of these entries can be found in the section for this command (page 304).

6.27 FTSHWLOG Display logging records

Note on usage

User group: FT user and FT administrator

Functional description

With the FTSHWLOG command, you can obtain information on all FT requests logged by openFT. An important prerequisite is that the FT administrator has switched on the FT logging function. The logging records are marked as FT or FTAC or ADM, enabling you to identify the type of logging record.

FT logging

The FT user can view all FT logging records which relate to his user ID.
The FT administrator can display all the FT logging records in his system.
In the case of inbound requests, openFT only writes a log record if the request has already reached the data transfer phase. If, for instance, it is not possible to find a file, no FT log record is written.If no options are specified, openFT outputs the most recent logging record. When requested, openFT outputs all the logging records within the time period specified in the command in reverse chronological order, i.e. starting from the most recent record to the oldest record.

Command execution may take several minutes, depending on the size of the log file!

There are three types of output: short output and long output and CSV format.

FTAC logging

With FTAC functionality, FTSHWLOG can be used to display the FTAC logging records. The FT user can view all FT logging records, of which he is the owner. FT and FTAC administrators may view all FT and FTAC logging records.

If the access check was positive and openFT accepted the request, a second logging record is created in openFT, indicating whether the request was completed successfully, and if not, why it was terminated.

Note

> It is no longer necessary for FT and FTAC administrators to cooperate in order to sure that logging data is archived in full (unlike in earlier versions).

A precise description of output can be found starting on page 323.

ADM logging

If your openFT instance is administered via a remote administration server or if you administer other instances yourself using FTADM, ADM log records are written (assuming that the appropriate logging settings have been made). You can also view these log records.

Format

(part 1 of 2)

```
FTSHWLOG

SELECT = *OWN / *ALL / *PARAMETERS(...)

  *PARAMETERS(...)
      LOGGING-ID = *ALL / <alphanum-name 1..12> / *INTERVAL(...)
        *INTERVAL(...)
            FROM = 1 / <alphanum-name 1..12>
           ,TO = *HIGHEST-EXISTING / <alphanum-name 1..12>
     ,OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
     ,CREATION-TIME = *INTERVAL(...) / *DAYS(...)
        *INTERVAL(...)
            FROM = 2000-01-01(...) / <date 8..10>(...)
              <date 8..10>(...)
                  TIME = 00:00 / <time 1..8>
           ,TO = *TOMORROW(...) / *TODAY(...) / <date 8..10>(...)
              <date 8..10>(...)
                  TIME = 00:00 / <time 1..8>
        *DAYS(...)
            NUMBER = <integer 1..1000>
     ,RECORD-TYPE = *ALL / *PARAMETERS(...)
        *PARAMETERS(...)
            FT = *TRANSFER-FILE / *NONE / list-poss(1): *TRANSFER-FILE
           ,FTAC = (*TRANSFER-FILE, *READ-FILE-ATTRIBUTES, *DELETE-FILE,
                  *CREATE-FILE, *MODIFY-FILE-ATTRIBUTES,
                  *READ-DIRECTORY, *MOVE-FILE, *CREATE-DIRECTORY,
                  *DELETE-DIRECTORY, *MODIFY-DIRECTORY. *LOGIN) / *NONE /
                  list-poss(11): *TRANSFER-FILE / *READ-FILE-ATTRIBUTES / *DELETE-FILE /
                  *CREATE-FILE / *MODIFY-FILE-ATTRIBUTES / *READ-DIRECTORY /
                  *MOVE-FILE / *CREATE-DIRECTORY / *DELETE-DIRECTORY /
                  *MODIFY-DIRECTORY / *LOGIN
           ,ADM = *ADMINISTRATION / *NONE / list-poss(1): *ADMINISTRATION
     ,INITIATOR = (*LOCAL, *REMOTE) / list-poss(2): *LOCAL / *REMOTE
     ,PARTNER = *ALL / <text 1..200 with-low>
     ,FILE-NAME = *ALL / <filename 1..59> / <filename-prefix 2..50> / <c-string 1..512 with-low> /
                 *DIRECTORY(...)
        *DIRECTORY(...)
            NAME = *ALL / <partial-filename 2..50> / <c-string 1..512 with-low>
     ,REASON-CODE = *ALL / *FAILURE / <text 1..4>
     ,ROUTING-INFO = *ALL / <text 1..200 with-low>
     ,TRANSFER-ID = *ALL / <integer 1.. 2147483639>
```

(part 2 of 2)

```
,NUMBER = 1 / *ALL / <integer 1..99999999>

,INFORMATION = *STD / *ALL

,OUTPUT = *STDERR(...) / *STDOUT(...)

   *STDERR(...) / *STDOUT(...)
     |  LAYOUT = *STD / *CSV
```

Operands

SELECT =
Selects a group of logging records.

SELECT = *OWN
Selects logging records under the user’s own login.

SELECT = *ALL
Displays all users’ logging records to the administrator.

SELECT = *PARAMETERS(...)

LOGGING-ID =
Number of the logging record.

LOGGING-ID = *ALL
The number of the logging record is not a selection criterion.

LOGGING-ID = <alphanum-name 1..12>
Number of the logging record to be output. The value range for the logging ID is from 1 through 999999999999.

LOGGING-ID = *INTERVAL(...)
Range of logging records to be output.

FROM = <alphanum-name 1..12>
First logging record to be output. The value range for the logging ID is from 1 through 999999999999.

TO = *HIGHEST-EXISTING / <alphanum-name 1..12>
Last logging record to be output. The value range for the logging ID is from 1 through 999999999999.

OWNER-IDENTIFICATION =
User ID whose logging records are to be displayed.

OWNER-IDENTIFICATION = *OWN
Logging records of your user ID are displayed.

OWNER-IDENTIFICATION = *ALL
The logging records of all user IDs are displayed. The FT administrator can thus display the FT logging records of any user ID, and the FTAC administrator can display the FTAC logging records of any user ID.

OWNER-IDENTIFICATION = <name 1..8>
Any user ID whose logging records should be displayed.

CREATION-TIME =
The range of the logging records to be output, selected by their date or time of creation.

CREATION-TIME = *INTERVAL(...)
The range is specified as a time interval using the date and/or time.

FROM = 2000-01-01(...) / <date 8..10>(...)
Date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2007-08-18 or 07-08-18 for 18 August, 2007. openFT then displays all logging records written after the specified date and time.

TIME = 00:00 / <time 1..8>
Time for the day specified with CREATION-TIME. openFT displays all logging records written after the specified time. The time is entered in the format *hh:mm:ss*, e.g. 14:30:10.

TO = *TOMORROW / *TODAY(...) / <date 8..10>(...)
Date in the format *yyyy-mm-dd* or *yy-mm-dd*, e.g. 2007-08-18 or 07-08-18 for 18 August, 2007. openFT then displays all logging records written up to the specified date and time.

TIME = 00:00 / <time 1..8>
Time for the day specified with CREATION-TIME. openFT displays all logging records written up to the specified time. The time is entered in the format *hh:mm:ss*, e.g. 14:30:10.

CREATION-TIME = *DAYS(NUMBER=<integer 1..1000>)
This field is specified in number of days. All logging sets that were created in the last n calendar days, including today, are output.

RECORD-TYPE =
Type of logging record to be displayed.

RECORD-TYPE = *ALL
The record type is not a selection criterion.

RECORD-TYPE = *PARAMETERS(...)
Type of the logging record.

FT = *TRANSFER-FILE / *NONE / list-poss(1): *TRANSFER-FILE
Specifies whether or not the FT logging records are to be displayed.

FTAC = (*TRANSFER-FILE, *READ-FILE-ATTRIBUTES, *DELETE-FILE, *CREATE-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY, *MOVE-FILE, *CREATE-DIRECTORY, *DELETE-DIRECTORY, *MODIFY-DIRECTORY, *LOGIN) / *NONE / list-poss(11): *TRANSFER-FILE / *READ-FILE-ATTRIBUTES / *DELETE-FILE / *CREATE-FILE / *MODIFY-FILE-ATTRIBUTES / *READ-DIRECTORY / *MOVE-FILE / *CREATE-DIRECTORY / *MODIFY-DIRECTORY / *DELETE-DIRECTORY / *LOGIN

Specifies whether or not FTAC logging records are to be displayed. If they are to be displayed, the FT function for which the FTAC logging records are to be displayed can also be specified. The following values are possible:

*TRANSFER-FILE
All logging records for the function “Transfer files” are displayed.

*READ-FILE-ATTRIBUTES
All logging records for the function “Read file attributes” are displayed.

*DELETE-FILE
All logging records for the function “Delete files” are displayed.

*CREATE-FILE
All logging records for the function “Create files” are displayed.

*MODIFY-FILE-ATTRIBUTES
All logging records for the function “Modify file attributes” are displayed.

*READ-DIRECTORY
All logging records for the function “Read file directory” are displayed.

*MOVE-FILE
All logging records for the function “Copy and delete files” are displayed.

*CREATE-DIRECTORY
All logging records for the function “Create directory” are displayed.

*DELETE-DIRECTORY
All logging records for the function “Delete directory” are displayed.

*MODIFY-DIRECTORY
All logging records for the function “Modify directory” are displayed.

*LOGIN
All logging records for the function “Inbound FTP access” are displayed. Log records of the type *LOGIN are only written in the case of an incorrect transfer admission.

ADM = *ADMINISTRATION / *NONE / list-poss(1): *ADMINISTRATION
Specifies whether ADM log records are output.

ADM = *ADMINISTRATION
ADM log records are output.

ADM = *NONE
No ADM log records are output.

INITIATOR =
Logging records according to the initiator.

INITIATOR = (*LOCAL,*REMOTE)
The initiator is not a selection criterion.

INITIATOR = *LOCAL
Only those logging records that belong to requests issued locally are displayed.

INITIATOR = *REMOTE
Only those logging records belonging to requests made from a remote system are displayed.

PARTNER =
The partner system.

PARTNER = *ALL
The partner system is not a selection criterion.

PARTNER = <text 1..200 with-low>
Name or address of the partner system for which the logging records are to be displayed. For more information on address specifications, see section “Specifying partner addresses” on page 108.

FILE-NAME =
File name.

FILE-NAME = *ALL
The file name is not a selection criterion.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Fully qualified name of the files for which you wish to view the logging records.

FILE-NAME = <filename-prefix 2..50>
Partially qualified name of the files for which you want to view the logging records.

Examples

- If you specify TOOLS as the beginning of the filename, all logging records containing the filename TOOLS.CLIST, TOOLS.CNTL or TOOLS.CLIST(MEMBER01) will be displayed.
- If you specify TOOLS.CLIST/ as the beginning of the filename, all logging records containing the filename TOOLS.CLIST(MEMBER01), TOOLS.CLIST(MEMBER02), etc. are displayed.

FILE-NAME = *DIRECTORY(...)
Name of the directory.

***DIRECTORY(...)**
The directory specification relates to the corresponding specification in the FTSHW command (see User Guide).

NAME = <u>*ALL</u>
The directory is not a selection criterion

NAME = <partial-filename 2..50> / <c-string 1..512 with-low>
Name of the directory.

Example

If you specify FILE=*DIR(NAME=ABC.) here, and not FILE=ABC., only those logging records are displayed that contain ABC (as the name of a file directory which were accessed from a remote system with the file management command in order to display an z/OS file directory).

REASON-CODE =
Selection by the reason code of the logging records.

REASON-CODE = <u>*ALL</u>
The reason code is not a selection criterion; all records are output.

REASON-CODE = *FAILURE
All logging records with error codes are output.

REASON-CODE = <text 1..4>
Logging records to be output by the error codes. Leading zeros can be omitted (e.g. 14 for FTR0014).

ROUTING-INFO = *ALL / <text 1..200 with-low>
Selects the ADM log records on the basis of the routing information. The routing information describes the administered instance in the case of remote administration requests issued locally.

ROUTING-INFO = <u>*ALL</u>
The routing information is not used as a selection criterion.

ROUTING-INFO = <text 1..200 with-low>
Routing information for which the ADM log records are to be output.

TRANSFER-ID =
Selection on the basis of the request ID.

TRANSFER-ID = *ALL
The request ID is not used as a selection criterion.

TRANSFER-ID = <integer 1..2147483639>
Only outputs log records for the specified request ID.

NUMBER =
Maximum number of logging records.

NUMBER = 1 / <integer 1..99999999>
The maximum number of logging records that are to be displayed. The default value is 1.

NUMBER = *ALL
All logging records are displayed.

INFORMATION =
Scope of the requested information.

INFORMATION = *STD
The logging records are displayed in a standard format (see page 320).

INFORMATION = *ALL
The logging records are displayed in a detailed format (see page 323).

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**omma **S**eparated **V**alue) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 181).

6.27.1 Description of the short output

Short output form of FT logging records (example)

```
FTSHWLOG NUMBER = 2

 TYP LOGG-ID  TIME      RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 2007-04-22
 T       5333 14:18:24 2169 <G133H301 FT2V292        FT2V292  TEST2
 T       5284 14:08:12 0000 >G133H301 FT2V292        FT2V292  TEST1
```

Short output format for ADM log records (examples)

ADM log record for a remote administration request that has been issued locally:

```
FTSHWLOG NUMBER=1

 TYP LOGG-ID TIME    RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 2009-06-03
 A     3555 17:15:11 0000 <ftadm:/* FTV520         FTV520
```

ADM log record on the administered openFT instance:

```
FTSHWLOG NUMBER=1

 2009-06-03
 TYP LOGG-ID TIME    RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
 A     9006 11:32:51 0000 >ftadm:/* *REMOTE        ftadmin
```

The value A is output for an ADM log record in the TYP column. It is also possible for an FTAC log record to be written. The FILENAME column is empty for ADM log records.

Explanation

Name	Explanation	
TYP (column 1)	Specifies if it is an FT or FTAC or ADM or FTP log record. T indicates the FT logging record, C indicates the FTAC logging record, A indicates the ADM logging record.	
TYP (columns 2-3)	Definition of FT function:	
	␣ V A D C M R CD MD DD L	transfer file transfer file and delete send file (only inbound possible) read file attributes delete file create file modify file attributes read directory create director modify directory delete directory login (inbound FTP access)
LOGG-ID	Number of the log record (up to twelve digits)	
TIME	Time when the logging record was written	
RC	Reason Code. Indicates if a request was successfully executed, or if not, why it was rejected or terminated. If an FT request is rejected for "FTAC reasons" (e.g. 0014), the exact reason behind the termination can be found in the FTAC logging record of the system that rejected the request. Further information on the reason code can be obtained using the FTHELP xxxx command.	
PARTNER	Provides information about the partner system. The output includes the symbolic name under which the system administrator has entered the partner system in the partner list. In the short form, the partner system name is given an identifier from which you can determine the request direction.	
	>	The request direction is to the partner system. This direction is specified for a – send request, i.e. the data is transferred to the partner – request to view remote file attributes – request to view remote directories
	<	The request direction is to the local system. This direction is specified for a – receive request, i.e.the data is transferred to the local system – request to modify remote file attributes – request to delete remote files
INITIATOR	Request initiator; if initiative is from remote system: *REMOTE	
INIT	The field is always empty in z/OS and is only output for reasons of compatibility.	

Name	Explanation
USER-ADM	User ID in the local system used by the requests
FILENAME	Filename resp. pre-processing or post-processing in the local system. In the case of ADM logging records, this field is empty. For security reasons, only the first 32 characters (or 42 characters in the case of FTEXECSV pre-processing operations) of a preprocessing or postprocessing command are taken over into the logging record. By arranging the call parameters accordingly or by inserting spaces, you can influence the command parameters that are not to appear in the logging record. FTEXECSV is the reaction to an ftexec command issued in a remote Windows or Unix system.

6.27.2 Description of the long output

Long output form (example)

```
LOGGING-ID  = 00003705  RC      = 2072        TIME     = 2007-04-19 13:43:11
   TRANS    = TO        REC-TYPE= FT          FUNCTION = TRANSFER-FILE
   PROFILE  =           PCMD    = NONE        STARTTIME=
   TRANS-ID = 69158     WRITE   = REPLACE     REQUESTED= 2007-04-19 13:41:32
   TRANSFER =         0 kB                    CCS-NAME = IBM1047
   INITIATOR= SYSXY
   USER-ADM = SYSXY
   PARTNER  = HUGO
   FILENAME = BULLETIN
LOGGING-ID  = 00003704  RC      = 0000        TIME     = 2007-04-19 13:41:32
   TRANS    = TO        REC-TYPE= FTAC        FUNCTION = TRANSFER-FILE
   PROFILE  =           PRIV    =
   INITIATOR= SYSXY
   USER-ADM = SYSXY
   PARTNER  = HUGO
   FILENAME = BULLETIN
```

Long output format for an ADM log record (example)

```
LOGGING-ID  = 1290      RC       = 0000      TIME     = 2009-01-17 08:24:46
   TRANS    = TO        REC-TYPE = ADM       FUNCTION = REM-ADMIN
   TRANS-ID  = 223905    PROFILE  = adminstr
   SEC-OPTS = ENCR+DICHK, RSA-768 / AES-128
   INITIATOR= *REMOTE
   USER-ADM = ftadmin8
   PARTNER  = ftadm://server.mch.mynet
   ADM-CMD  = FTCREPRF
   ADMIN-ID =
   ROUTING  =
```

Explanation of long output form (column-wise)

Name	Explanation	
LOGGING-ID	Number of the log record (up to twelve digits)	
TRANS	Transfer direction:	
	TO	The request direction is to the partner system. This direction is specified for a – send request, i.e. the data is transferred to the partner. – request to view remote file attributes – request to view remote directories
	FROM	The request direction is to the local system (inbound). This direction is specified for a – receive request, i.e. the data are transferred to the local system – request to modify remote file attributes – request to delete remote files
	BOTH	File management request with two-way data transfer.
PROFILE	Name of the profile to be used for the transfer (empty in the FT logging record)	
TRANS-ID	Transfer ID number	
TRANSFER	Amount of data transferred	
SEC-OPTS	Security options and encryption algorithms used. This line is only output if at least one of the options is used.	
	ENCR	Encryption of the request queue
	DICHK	Data integrity check of the request queue
	DENCR	Encryption of data content during the transfer
	DDICHK	Data integrity check of the file data to be transferred
	LAUTH	Authentication of the local system on a partner
	RAUTH	Authentication of the partner on a local system
	RSA-nnnn	Length of the RSA key
	DES / AES	Encryption algorithm used
INITIATOR	Request initiator; if initiative is from remote system: *REMOTE	
USER-ADM	User ID in the local system used by the requests	
PARTNER	Provides information about the partner system. The output includes the symbolic name under which the system administrator has entered the partner system in the partner list.	

Name	Explanation	
FILENAME	Filename resp. pre-processing or post-processing in local system. For security reasons, only the first 32 characters (or 42 characters in the case of FTEXECSV pre-processing operations) of a preprocessing or postprocessing command are taken over into the logging record. By arranging the call parameters accordingly or by inserting spaces, you can influence the command parameters that are not to appear in the logging record. FTEXECSV is the reaction to an ftexec command issued in a remote Windows or Unix system.	
ADM-CMD	Only output for an ADM log record: Administration command without parameters	
ADMIN-ID	Only output for an ADM log record:: Remains empty because only relevant on the remote adminsitration server	
ROUTING	Only output for an ADM log record:: Routing information on the openFT instance to be administered	
RC	Reason-Code. Indicates if a request was successfully executed, or if not, why it was rejected or terminated. If an FT request is rejected for “FTAC reasons” (e.g. 2169), the exact reason behind the termination can be found in the FTAC logging record of the system that rejected the request. Further information on the reason code can be obtained using the FTHELP xxxx command.	
REC-TYPE	Specifies if this is an FT or FTAC or ADM logging record.	
PCMD	Status of follow-up processing:	
	NONE	No follow-up processing defined.
	STARTED	Follow-up processing was started.
	NOT-STARTED	Follow-up could not be started.
PRIV	specifies whether the admission profile is privileged.	
WRITE	Write rules:	
	NEW	A new file is created. If a file with the same name already exists, the transfer will be aborted.
	EXT	An existing file is extended and stored as new.
	REPLACE	An existing file is extended.
TIME	Time when the logging record was written	

Name	Explanation
FUNCTION	Definition of FT function:
	– TRANSFER-FILE: transfer file – MOVE-FILE: transfer file and delete send file (only inbound possible) – READ-FILE-ATTRIBUTES: read file attributes – DELETE-FILE: delete file – CREATE-FILE: create new file – MODIFY-FILE-ATTRIBUTES: modify file attributes – READ-DIRECTORY: read directory – CREATE-DIRECTORY: create directory – MODIFY-DIRECTORY: modify directory – DELETE-DIRECTORY: delete directory – LOGIN: inbound FTP access – REM-ADMIN: remote administrator
STARTTIME	Time request was started
STORETIME	Time request was accepted (inbound)
REQUESTED	Time request was accepted (outbound)
CCS-NAME	Name of the character set, used for code conversion as necessary.
INITSN	TSN from which the request came. If the INITIATOR was *REMOTE, the field is empty.

Example 1

The FT administrator wants to display all logging records that were created for the user ID *Meier* and logged between 01.01.2009 and 31.03.2009. If you are the owner of the User ID

```
FTSHWLOG SELECT=*PARAMETERS(OWNER-IDENTIFICATION=MEIER,          -
                CREATION-TIME=*INTERVAL(FROM=2009-01-01(00:00), -
                TO=2009-03-31(23:59))),NUMBER=*ALL
```

You want to see the first record of the output in detail.

```
FTSHWLOG (OWN=Meier,CRE-TIME=*INTERVAL(FROM=2009-01-01(00:00),  -
          TO=2009-03-31(00:00))),INF=*ALL
```

Example 2

An (FT or FTAC) administrator wants to view all logging records. He wants all the information to be output in the most compact possible form because he wants to back up the logging records before deleting them. To do this, he combines the specifications for "comprehensive output" and "output in CSV format". This is achieved using the following command:

```
FTSHWLOG SELECT=*ALL,NUMBER=*ALL,INF=*ALL,OUTPUT=*STDOUT(*CSV)
```

This command also takes a few minutes to output comprehensive information.

6.28 FTSHWMON Show monitoring data

Note on usage

User group: FT users and FT administrators

Description of the function

The FTSHWMON command allows you to output the monitoring values from openFT operation on the local system. To do this, monitoring must be activated (see FTMODOPT) and openFT must be activated.

Format

```
FTSHWMON
,NAME = *STD / *ALL /<list-poss(100): alphanum-name 1..12>
,POLLING =*NONE / *PARAMETERS(...)
   *PARAMETERS(...)
      | INTERVAL= 1 /<integer 1..600>
      | ,NUMBER = *UNLIMITED / <integer 1..3600>
,INFORMATION = *VALUES(...) / *TYPE
   *VALUES(...)
      | DATA = *FORMATTED / *RAW
,OUTPUT = *STDERR(...) /  *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
      |  LAYOUT = *STD / *CSV
```

Operands

NAME =
Specifies what monitoring values are to be output.

NAME = *STD
A predefined default set of monitoring values is output. See “Examples” on page 335.

NAME = *ALL
All monitoring values are output.

NAME = <list-poss(100): alphanum-name 1..12>
Here you can enter a list of up to 100 names of monitoring values that are to be output. The name must be one of the short names (see the table in the section "Description of the monitoring values" on page 330).

POLLING =
Specifies the interval at which the monitoring values are to be polled.

POLLING =<u>*NONE</u>
The monitoring values are only polled once.

POLLING =*PARAMETERS
In this structure you specify a time interval and a repetition factor for polling the monitoring values. If an error occurs during polling, further repeated output is canceled.

> **INTERVAL = <u>1</u>**
> The time interval for polling the monitoring values is 1 second.
>
> **INTERVAL = <integer 1..600>**
> Time interval in seconds for polling the monitoring values.
>
> **NUMBER = <u>*UNLIMITED</u>**
> There is no limit to the number of times the monitoring values are polled. To cancel the command, you can use the key combination PA1 and RESET, for example.
>
> **NUMBER = <integer 1..3600>**
> Here you specify how often the monitoring values are to be polled.

INFORMATION =
Specifies whether the monitoring values themselves or the type of the monitoring values is to be output.

INFORMATION = <u>*VALUES(...)</u>
The measured value is output. You can specify whether the monitoring values are to be output in formatted form or as raw data.

> **DATA =<u>*FORMATTED</u>**
> The monitoring values are formatted for visual display, e.g. as throughput, maximum or average.
>
> **DATA =*RAW**
> Raw, unformatted data is output. Monitoring values for the duration of an action are not output.

INFORMATION = *TYPE
Outputs the type and, where applicable, the scaling factor of the monitoring value or the type of the metadata.

The scaling factor is only of significance for some monitoring values and in CSV format if *RAW is not specified. In this case, the output value must be divided by the scaling factor to get the real value. In the case of formatted data in tabular format, the scaling factor 100 specifies that the number is output to 2 decimal places.

The following output values are possible for *TYPE:

*BOOL	Boolean value
*PERCENT	Percentage
*INT	Integer number (corresponds to *INT(1))
*INT(100)	Integer value with a scaling factor of 100
*TIME	Timestamp
*STRING	Text output for the selection

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
The data is output to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
The data is output to SYSPRINT.

LAYOUT = *STD
Output is formatted in a form readable by the user.
If the monitoring configuration changes (filters), a new header and a new start time for monitoring is output in standard output format.

LAYOUT = *CSV
Data is output in **C**omma **S**eparated **V**alue format. This is a quasi-tabular format that is in widespread use in the field of PCs and in which the individual fields are separated by semicolons ";" (see section “Output in CSV format” on page 181).

If the monitoring configuration changes (filters), the new start time for monitoring is shown in a separate column in CSV format.

6.28.1 Description of the monitoring values

The table below shows all the monitoring values output when NAME=*ALL is specified. Under NAME=, you can also specify a list of any of the parameters shown in the table.

The first two letters of the name indicate the data object that the monitoring value belongs to.

- Th = Throughput
- Du = Duration
- St = State

The second component of the name indicates the performance indicator, e.g. Netb for net bytes. In the case of monitoring values for the Throughput or Duration data object, the last 3 letters of the name indicate the types of requests from which the monitoring value originates, e.g.

- Ttl = FT Total
- Snd = FT Send requests
- Rcv = FT Receive requests
- Txt = Transfer of text files
- Bin = Transfer of binary files
- Out = FT Outbound
- Inb = FT Inbound

i If monitoring is deactivated for all partners (PARTNER-SELECTION=*NONE with FTMODOPT ...,MONITORING), only the following values are provided:

Status: StCLim, StCAct, StRqLim, StRqAct, StOftr, StFtmr, StFtpr, StTrcr

All the other values are set to 0.

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
ThNetbTtl	Throughput in net bytes: Number of bytes transferred	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbSnd	Throughput in net bytes (send requests): Number of bytes transferred with send requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbRcv	Throughput in net bytes (receive requests): Number of bytes transferred with receive requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThNetbTxt	Throughput in net bytes (text files): Number of bytes transferred when transferring text files	*ALL	Number of bytes per second	Bytes, accumulated

Name	Meaning	Output with	Output unit	
			FORMATTED	**RAW**
ThNetbBin	Throughput in net bytes (binary files): Number of bytes transferred when transferring binary files	*ALL	Number of bytes per second	Bytes, accumulated
ThDiskTtl	Throughput in disk bytes: Number of bytes read from files or written to files with transfer requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskSnd	Throughput in disk bytes (send requests): Number of bytes read from files with send requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskRcv	Throughput in disk bytes (receive requests): Number of bytes written to files with receive requests	*STD/ *ALL	Number of bytes per second	Bytes, accumulated
ThDiskTxt	Throughput in disk bytes (text files): Number of bytes read from text files or written to text files with transfer requests	*ALL	Number of bytes per second	Bytes, accumulated
ThDiskBin	Throughput in disk bytes (binary files): Number of bytes read from binary files or written to binary files with transfer requests	*ALL	Number of bytes per second	Bytes, accumulated
ThRqto	openFT requests: Number of openFT requests received	*STD/ *ALL	Number per second	Accumulated number
ThRqft	File transfer requests: Number of file transfer requests received	*ALL	Number per second	Accumulated number
ThRqfm	File management requests: Number of file management requests received	*ALL	Number per second	Accumulated number
ThSuct	Successful requests: Number of successfully completed openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThAbrt	Aborted requests: Number of aborted openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThIntr	Interrupted requests: Number of interrupted openFT requests	*STD/ *ALL	Number per second	Accumulated number
ThUsrf	Requests from non-authorized users: Number of openFT requests in which the user check was terminated with errors	*STD/ *ALL	Number per second	Accumulated number

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
ThFoll	Started follow-up processing operations: Number of follow-up processing operations started	*ALL	Number per second	Accumulated number
ThCosu	Connections established: Number of connections successfully established	*ALL	Number per second	Accumulated number
ThCofl	Failed connection attempts: Number of attempts to establish a connection that failed with errors	*STD/ *ALL	Number per second	Accumulated number
ThCobr	Disconnections: Number of disconnections as a result of connection errors	*STD/ *ALL	Number per second	Accumulated number
DuRqtlOut	Maximum outbound request duration: Maximum request duration of an outbound request	*ALL	Milliseconds [1]	-
DuRqtlInb	Maximum inbound request duration: Maximum request duration of an inbound request	*ALL	Milliseconds [1]	-
DuRqftOut	Maximum outbound transfer request duration: Maximum duration of an outbound file transfer request	*ALL	Milliseconds [1]	-
DuRqftInb	Maximum inbound transfer request duration: Maximum duration of an inbound file transfer request	*ALL	Milliseconds [1]	-
DuRqfmOut	Maximum outbound file management request duration: Maximum duration of an outbound file management request	*ALL	Milliseconds [1]	-
DuRqfmInb	Maximum inbound file management request duration: Maximum duration of an inbound file management request	*ALL	Milliseconds [1]	-
DuRqesOut	Maximum outbound request waiting time: Maximum waiting time before an outbound request is processed (for requests without a specific start time)	*ALL	Milliseconds [1]	-

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
DuDnscOut	Maximum duration of an outbound DNS request Maximum time an outbound openFT request was waiting for partner checking	*ALL	Milliseconds [1]	-
DuDnscInb	Maximum duration of an inbound DNS request Maximum time an inbound openFT request was waiting for partner checking	*ALL	Milliseconds [1]	-
DuConnOut	Maximum duration of establishment of a connection: Maximum time between requesting a connection and receiving confirmation of a connection for an outbound openFT request	*ALL	Milliseconds [1]	-
DuOpenOut	Maximum file open time (outbound): Maximum time an outbound openFT request required to open the local file	*ALL	Milliseconds [1]	-
DuOpenInb	Maximum file open time (inbound): Maximum time an inbound openFT request required to open the local file	*ALL	Milliseconds [1]	-
DuClosOut	Maximum file close time (outbound): Maximum time an outbound openFT request required to close the local file	*ALL	Milliseconds [1]	-
DuClosInb	Maximum file close time (inbound): Maximum time an inbound openFT request required to close the local file	*ALL	Milliseconds [1]	-
DuUsrcOut	Maximum user check time (outbound): Maximum time an outbound openFT request required to check the user ID and transfer admission	*ALL	Milliseconds [1]	-
DuUsrcInb	Maximum user check time (inbound): Maximum time an inbound openFT request required to check the user ID and transfer admission	*ALL	Milliseconds [1]	-
StRqas	Number of synchronous requests in the ACTIVE state	*STD/ *ALL	Average [2]	Current number
StRqaa	Number of asynchronous requests in the ACTIVE state	*STD/ *ALL	Average value [2]	Current number
StRqwt	Number of requests in the WAIT state	*STD/ *ALL	Average value [2]	Current number

Name	Meaning	Output with	Output unit	
			FORMATTED	RAW
StRqhd	Number of requests in the HOLD state	*STD/ *ALL	Average value [2]	Current number
StRqsp	Number of requests in the SUSPEND state	*STD/ *ALL	Average value [2]	Current number
StRqlk	Number of requests in the LOCKED state	*STD/ *ALL	Average value [2]	Current number
StRqfi	Number of requests in the FINISHED state	*ALL	Average value [2]	Current number
StCLim	Maximum number of connections: Upper limit for the number of connections established for asynchronous requests.	*STD/ *ALL	Value currently set	
StCAct	Number of occupied connections for asynchronous requests	*STD/ *ALL	Share of StCLim in % [3]	Current number
StRqLim	Maximum number of requests: Maximum number of asynchronous requests in request management	*STD/ *ALL	Value currently set	
StRqAct	Entries occupied in request management	*STD/ *ALL	Share of StRqLim in % [3]	Current number
StOftr	openFT protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StFtmr	FTAM protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StFtpr	FTP protocol activated/deactivated	*STD/ *ALL	ON (activated) OFF (deactivated)	
StTrcr	Trace activated/deactivated	*ALL	ON (activated) OFF (deactivated)	

[1] Maximum value during the last monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring). The minimum time interval output is 1 millisecond if a relevant measurement has been completed during the interval since the last query. A value of 0 specifies that no measurement has been made in this interval.

[2] Average value during the monitoring interval (= time elapsed since the last time the monitoring values were queried or since the start of monitoring). The format is n.mm, where n is an integer and mm are to be interpreted as decimal places.

[3] If the reference value is reduced in live operation, it is possible for the value output to lie above 100 (%) temporarily.

6.28.2 Examples

1. Monitoring values are to be output in default output format.

```
FTSHWMON
openFT(STD)   Monitoring (formatted)
MonOn=2009-02-16 15:36:12 PartnerSel=OPENFT  RequestSel=ONLY-ASYNC,ONLY-LOCAL
2009-02-17 15:40:01

 Name      Value
 ----------------
 ThNetbTtl 38728
 ThNetbSnd 38728
 ThNetbRcv 0
 ThDiskTtl 16384
 ThDiskSnd 16384
 ThDiskRcv 0
 ThRqto    1
 ThSuct    0
 ThAbrt    0
 ThIntr    0
 ThUsrf    0
 ThCofl    0
 ThCobr    0
 StRqas    0.00
 StRqaa    8.66
 StRqwt    1.66
 StRqhd    0.00
 StRqsp    0.00
 StRqlk    0.00
 StCLim    16
 StCAct    37
 StRqLim   1000
 StRqAct   1
 StOftr    ON
 StFtmr    OFF
 StFtpr    OFF
```

Explanation

The default output format begins with a header containing the following specifications:

– Name of the openFT instance and selected data format (raw or formatted)

– Monitoring start time and partner and request selection

– Current timestamp

This is followed by the list of default values. See the section “Description of the monitoring values” on page 330 for the meanings.

2. Only the data types are to be output in default output format.

```
FTSHWMON INFORMATION=*TYPE
openFT(STD)   Monitoring (formatted)
MonOn=2009-02-16 15:36:12 PartnerSel=OPENFT  RequestSel=ONLY-ASYNC,ONLY-REMOTE
2009-02-17 15:40:01

 Name      Value
 ----------------
 ThNetbTtl INT
 ThNetbSnd INT
 ThNetbRcv INT
 ThDiskTtl INT
 ThDiskSnd INT
 ThDiskRcv INT
 ThRqto    INT
 ThSuct    INT
 ThAbrt    INT
 ThIntr    INT
 ThUsrf    INT
 ThCofl    INT
 ThCobr    INT
 StRqas    INT(100)
 StRqaa    INT(100)
 StRqwt    INT(100)
 StRqhd    INT(100)
 StRqsp    INT(100)
 StRqlk    INT(100)
 StCLim    INT
 StCAct    PERCENT
 StRqLim   INT
 StRqAct   PERCENT
 StOftr    BOOL
 StFtmr    BOOL
 StFtpr    BOOL
```

Explanation

The types in the Value column have the following significance:

INT	Integer number (corresponds to INT(1))
INT(100)	Numeric value with a scaling value of 100 in the format n.mm, where n is an integer and mm are decimal places.
PERCENT	Percentage
BOOL	Boolean value, ON / OFF

3. The monitoring value "throughput in netbytes" (ThNetbTtl) is to be displayed. The display is to be updated every 60 seconds and repeated three times (polling).

   ```
   FTSHWMON NAME=ThNetbTtl,POLLING=*PAR(INTERVAL=60,NUMBER=3)
   ```

   ```
   openFT(STD)   Monitoring (formatted)
   MonOn=2009-02-19 10:44:09 PartnerSel=OPENFT,FTP  RequestSel=ONLY-ASYNC,ONLY-LOCAL
   2009-02-19 12:45:33
    Name      Value
    ----------------
    ThNetbTtl 780107

   2009-02-19 12:46:33
    ThNetbTtl 993051

   2009-02-19 12:47:33
    ThNetbTtl 1049832
   ```

   The repetitions are separated by intermediate header containing the current polling time.

4. All values are to be output in CSV format.

   ```
   FTSHWMON NAME=*ALL,OUTPUT=*STDERR(*CSV)
   ```

   ```
   CurrTime;MonOn;PartnerSel;ReqSel;Data;ThNetbTtl;ThNetbSnd;ThNetbRcv;ThNetb
   Txt;ThNetbBin;ThDiskTtl;ThDiskSnd;ThDiskRcv;ThDiskTxt;ThDiskBin;ThRqto;ThR
   qft;ThRqfm;ThSuct;ThAbrt;ThIntr;ThUsrf;ThFoll;ThCosu;ThCofl;ThCobr;DuRqtlO
   ut;DuRqtlInb;DuRqftOut;DuRqftInb;DuRqfmOut;DuRqfmInb;DuRqesOut;DuDnscOut;D
   uDnscInb;DuConnOut;DuOpenOut;DuOpenInb;DuClosOut;DuClosInb;DuUsrcOut;DuUsr
   cInb;StRqas;StRqaa;StRqwt;StRqhd;StRqsp;StRqlk;StRqfi;StCLim;StCAct;StRqLi
   m;StRqAct;StOftr;StFtmr;StFtpr;StTrcr
   2009-08-13 10:44:24;2009-08-13 10:35:46;*ALL;*ALL;FORM;0;0;0;0;0;0;0;0;0;
   0;0;0;0;0;0;0;0;0;0;0;0;0;5129;0;5129;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
   0;16;0;2000;0;1;0;1;0
   ```

6.29 FTSHWNET Display the network environment

Note on usage

User group: FT administrator

This command has to be entered under TSO.

Functional description

You use this command to output information about the network environment of the current openFT instance.

Format

FTSHWNET

Without operands

Example

```
ftshwnet
openFT: VERSION    = 11.0A00
openFT: IP-ADDR    = 111.22.123.34
openFT: PORT-NR    = 1100
openFT: TCP-NAME   = TCPIP
openFT: INSTANCE   = STD
openFT: VTAM-FTID  = PBFT2
openFT: CMD-TRANS  = TCP
openFT: MSG-CRYPT  = N
openFT: SVC IN USE = 211
openFT: SS  IN USE = OPFT
```

6.30 FTSHWOPT Display operating parameters

Note on usage

User group: FT user and FT administrator

Functional description

The command FTSHWOPT can be used at any time to obtain the information listed below on the operating parameters of your FT system:

- Maximum number of file transfer requests in the request queue,
- Maximum number of tasks that can process file transfer requests concurrently,
- Maximum number of transport connections that can be reserved simultaneously for file transfer requests,
- Maximum size of a transport unit in bytes,
- Default setting for the FTAC security level of the partner systems,
- Status of extended authentication check (on or off),
- Which type of trace is switched on,
- Whether the logging is switched on,
- Maximum lifetime of FT requests in the request file,
- The instance ID.
  After installation, this field is supplied with the name of the real host.
- Whether and which console traps are activated or deactivated.
- Whether and which ADM traps are activated or deactivated and to which ADM trap server the activated traps are sent. The transfer admission of the ADM trap server is not output with the default output format and CSV output format. It does, however, appear in the output as a command string (*BS2-PROC, *ZOS-PROC)!
- Whether and which monitoring functions are activated.

Format

```
FTSHWOPT
,OUTPUT = *STDERR(...) / *STDOUT(...)

   *STDERR(...) / *STDOUT(...)
     | LAYOUT = *STD / *CSV / *BS2-PROC / *ZOS-PROC
```

Operands

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is put into a user-friendly form for reading.

LAYOUT = *CSV
Output takes place in **C**omma **S**eparated **V**alue format. This is a special tabular format, widely used in the PC world, where the individual fields are separated by semicolons “;“ (see section “Output in CSV format” on page 181).

LAYOUT = *BS2-PROC
The operating parameters are output as a command sequence. This can be called as an SDF procedure at BS2000/OSD systems in order to recreate the identical operating parameters.

LAYOUT = *ZOS-PROC
The operating parameters are output as a command sequence. This can be called as a Clist procedure at z/OS systems in order to recreate the identical operating parameters.

Example

Default of the FTSHWOPT command, i.e. the operating parameters have not been modified since installation.

```
FTSHWOPT
STARTED PROC-LIM CONN-LIM ADM-CLIM RQ-LIM MAX-RQ-LIFE TU-SIZE KEY-LEN CCS-NAME
    YES      2       16       8     2000      30       65535    768   IBM1047
PTN-CHK DYN-PART SEC-LEV  FTAC-LOG FT-LOG ADM-LOG
   STD      ON   B-P-ATTR    ALL    ALL    ALL
OPENFT-APPL      FTAM-APPL         FTP-PORT    ADM-PORT
*STD             *NONE             21          11000
ACTIVE           NAVAIL            ACTIVE      ACTIVE
HOST-NAME        IDENTIFICATION / LOCAL SYSTEM NAME
*NONE            FJMFTFOA / $FJAM,FJMFTFOA

ADM-TRAP-SERVER
*NONE

TRAP: SS-STATE FT-STATE PART-STATE PART-UNREA RQ-STATE TRANS-SUCC TRANS-FAIL
CONS    OFF      OFF       OFF        OFF       OFF       OFF        OFF
ADM              OFF       OFF        OFF       OFF       OFF        OFF

FUNCT:  SWITCH   PARTNER-SELECTION   REQUEST-SELECTION   OPTIONS
MONITOR OFF     ALL                 ALL
TRACE   OFF     ALL                 ALL                 NONE
```

Meaning of the output fields

STARTED
Specifies whether openFT is activated (via FTSTART or automatically) or not.

PROC-LIM
Maximum number of tasks that can be reserved simultaneously for the execution of FT requests. The value is defined by the PROCESS-LIMIT operand in the FTMODOPT command.
Default value: 2

CONN-LIM
Maximum number of transport connections that can be reserved for asynchronous file transfer requests. Since each transport connection can only process one request at a time, CONN-LIMIT also defines the maximum number of requests that can be processed simultaneously. One third of the transport connections are reserved for requests from remote systems. The value of CONN-LIMIT is defined by the CONN-LIMIT operand in the FTMODOPT command.
Default value: 16

ADM-CLIM
Maximum number of asynchronous administration requests including ADM traps that can be processed simultaneously. The value of ADM-CLIM is specified with the operand ADM-CONN-LIM in the command FTMODOPT.
Default value: 8

RQ-LIM
Maximum number of FT requests that can be entered at the same time in the request queue of the local system. The value can be modified using the REQUEST-LIMIT operand in the FTMODOPT command.
Default value: 2000

MAX-RQ-LIFE
Maximum number of days that an FT request is stored in the request file after its start time. When this period expires, the FT request is automatically removed from the request file. The value is defined in the MAX-REQUEST-LIFETIME operand of the FTMODOPT command.
Default value: 30

TU-SIZE
Maximum size of a transport unit in bytes. The value is defined with the TRANSPORT-UNIT-SIZE operand in the FTMODOPT command. The load placed on the transport system by openFT can be controlled using this operand.
Default value: 65535

KEY-LEN
Current length of the RSA key. 0 means that encryption is deactivated. The value is defined with the KEY-LENGTH operand in the FTMODOPT command.

CCS-NAME
Name of the character set, which is used as standard character set for FT requests. The standard character set can be created with the CODED-CHARACTER-SET operand of the FTMODOPT command.
Default value: IBM1047

PTN-CHK
Defines whether or not enhanced sender checking is activated. The value is defined with the PARTNER-CHECK operand in the FTMODOPT command.
Default value: STD

DYN-PART
specifies whether dynamic partners are permitted (*ON) or not (*OFF). The value is defined with the DYNAMIC-PARTNERS operand in the FTMODOPT command.
Default value: ON

SEC-LEV
Local default value for the security level of the partner systems. This operand is only effective if FTAC functionality is being used. An important part of the access protection functions provided by this product lies in the allocation of security levels to remote systems. To this end, each system is allocated a security level designated using an integer in the range 1 to 100. A default value is set by using the SECURITY-LEVEL operand in the FTMODOPT command. This value can then be defined for an individual system by specifying SECURITY-LEVEL=STD in the FTADDPTN command.
Default value for partners in the partner list: *STD (the security level depends on the type of partner).
In the case of dynamic partners, the value *BY-PARTNER-ATTRIBUTES is always output.

FTAC-LOG
Scope for FTAC logging (ALL, MODIFY, REJECTED).

FT-LOG
Scope for FT logging (ALL, FAIL, NONE).

ADM-LOG
Scope of ADM logging (ALL, FAIL, MODIFY, NONE).
The scope of FT, FTAC and ADM logging is specified in the FTMODOPT command.

OPENFT-APPL
Port number used by the local openFT. *STD means that the default port number 1100 is used. The value is specified with the OPENFT-APPLICATION operand in the command FTMODOPT.
The second line specifies whether the asynchronous inbound server is activated for openFT (ACTIVE), deactivated (DISABLED) or unavailable (INACT). The ACTIVE-APPLICATIONS operand in the command FTMODOPT is used for activation and deactivation.

FTAM-APPL
Not relevant on z/OS systems; is always supplied with *NONE.

FTP-PORT
Port number used by the local FTP server. The default value is 21. The value is specified with the FTP-PORT operand in the command FTMODOPT.
The second line specifies whether the asynchronous inbound server is activated for FTP (ACTIVE/DISABLED) or is unavailable or not installed (INACT/NAVAIL). The ACTIVE-APPLICATIONS operand in the command FTMODOPT is used for activation and deactivation.

ADM-PORT
Specifies the port number used by the local FT for remote administration. The default value is 11000. The value is specified with the ADM-PORT operand in the command FTMODOPT. The second line specifies whether the asynchronous inbound server is activated for remote administration requests (ACTIVE), deactivated (DISABLED) or unavailable (INACT). The ACTIVE-APPLICATIONS operand in the command FTMODOPT is used for activation and deactivation.

HOST-NAME
Name of the host that is automatically taken over if you have specified a host during the FJGEN initialization run.

IDENTIFICATION / LOCAL SYSTEM NAME
Instance identifier of the openFT instance currently set and the name of the local system. The instance identifier is defined with the IDENTIFICATION operand of the FTMODOPT command and is used to identify the instance in the partner systems.

ADM-TRAP-SERVER
Name or address of the partner to which the ADM traps are sent.
*NONE means that the sending of ADM traps is deactivated.
The value is specified with the ADM-TRAPS=(DESTINATION=...) operand in the command FTMODOPT.

TRAP
This section with the rows CONS and ADM specifies the trap settings. The columns identify the events for which traps may be generated.

- SS-STATE: Subsystem state change
- FT-STATE: State change of the openFT control process
- PART-STATE: Partner system state change
- PART-UNREA: Partner not reachable
- RQ-STATE: Request management state change
- TRANS-SUCC: Successfully completed requests
- TRANS-FAIL: Failed requests

The possible values are ON or OFF.
Default value: OFF (for all columns)

The following rows specify the settings for the various trap types:

CONS
Settings for console traps FTR03XXX. This is specified with the CONSOLE-TRAPS operand in the command FTMODOPT.

ADM
Setting for ADM traps to be output to the ADM trap server. This is specified with the ADM-TRAPS=(SELECTION=...) operand in the command FTMODOPT.

FUNCT
This section specifies the settings for monitoring (MONITOR) and tracing (TRACE). The columns have the following meanings:

- SWITCH: Function activated (ON) or deactivated OFF
- PARTNER-SELECTION: Selection according to protocol type of the partner system: ALL, OPENFT, FTP, ADM (only with TRACE), NONE
- REQUEST-SELECTION: Selection according to request type: ALL, ONLY-ASYNC, ONLY-SYNC, ONLY-LOCAL, ONLY-REMOTE
- OPTIONS (only with TRACE): NONE, NO-BULK-DATA (= minimal trace, i.e. no bulk data)

The following rows specify what the settings apply to:

MONITOR
Setting for monitoring. This is specified with the MONITORING operand in the command FTMODOPT.

TRACE
Setting for trace function. This is specified with the TRACE operand in the command FTMODOPT.

6.31 FTSHWPRF Display admission profile

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

With the command FTSHWPRF, FTAC users can obtain information about their admission profiles. The FTAC administrator can obtain information about all the admission profiles in his system.

Either the contents of the selected admission profile or only its name can be output. It is not possible to use FTSHWPRF to access defined passwords or transfer admissions defined in the profile! If a transfer admission is forgotten, a new one must be specified using FTMODPRF.

Format

```
FTSHWPRF

 NAME = *ALL / <alphanum-name 1..8> / *STD
,SELECT-PARAMETER = *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
    |   TRANSFER-ADMISSION = *ALL / *NOT-SPECIFIED / <alphanum-name 8..32> /
    |                                           <c-string 8..32 with-low> / <x-string 15..64>
    |  ,OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
,INFORMATION = *ONLY-NAMES / *ALL
,OUTPUT = *STDERR(...) / *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
    |   LAYOUT = *STD / *CSV
```

Operands

NAME =
Name of the admission profile you wish to view. NAME accesses the user-wide unique admission profile name.

NAME = *ALL
Views all admission profiles.

NAME = <alphanum-name 1..8>
Views the admission profile with the specified name.

NAME = *STD
Displays the default admission profile for your own user ID.

SELECT-PARAMETER =
Selection criteria for the admission profiles you wish to view.

SELECT-PARAMETER = *OWN
Views all the admission profiles of which you are the owner. This means that you can view all the admission profiles which are assigned to your user ID.

SELECT-PARAMETER = *PARAMETERS(...)
Selection criteria with which you can access your admission profiles.

TRANSFER-ADMISSION =
Transfer admission defined in an admission profile as a selection criterion.

TRANSFER-ADMISSION = *ALL
TRANSFER-ADMISSION is not used as a selection criterion.

TRANSFER-ADMISSION = *NOT-SPECIFIED
Only admission profiles for which no transfer admission has been specified are displayed.

TRANSFER-ADMISSION = <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64>
Views the admission profile which can be addressed with this transfer admission.

OWNER-IDENTIFICATION =
Specifies, whose admission profiles you wish to view.

OWNER-IDENTIFICATION = *OWN
Views only your own admission profile.

OWNER-IDENTIFICATION = *ALL
The FTAC administrator can view all admission profiles, regardless of who the owner is.

OWNER-IDENTIFICATION = <name 1..8>
The FTAC user can only access his own admission profiles; the output corresponds to *OWN. The FTAC administrator can view the admission profiles of any FTAC user with this parameter.

INFORMATION =
Scope of information desired.

INFORMATION = *ONLY-NAMES
FTAC only outputs the name of the admission profile and indicates whether it is privileged or blocked. An “*“ is output for privileged profiles and a “!“ for blocked profiles.

INFORMATION = *ALL
FTAC outputs the contents of the admission profile, excluding any passwords and the transfer admission.
In the case of a blocked admission profile (marked with an "!" when output with INFORMATION=*ONLY-NAMES), the following values can appear in TRANS-ADM:

TRANS-ADM=	Meaning
(NOT-SPECIFIED)	No TRANSFER-ADMISSION specified in the admission profile.
(DUPLICATED)	The admission profile was blocked because the TRANSFER-ADMISSION was "detected" by another user and the profile was to be blocked in that case (USAGE=*PRIVATE is specified in the command FTCREPRF or FTMODPRF). "Detected" means that another user ID tried to assign the same TRANSFER-ADMISSION over again.
(LOCKED(by_user))	The admission profile was specifically blocked by the user (VALID=*NO was specified in the command FTCREPRF or FTMODPRF).
(LOCKED(by_adm))	The admission profile was specifically blocked by the FTAC administrator (VALID=*NO was specified in the command FTCREPRF or FTMODPRF).
(EXPIRED)	The validity of TRANSFER-ADMISSION has expired (EXPIRATION-DATE was specified in command FTCREPRF or FTMODPRF).

OUTPUT =
Output medium for the information.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**omma **S**eparated **V**alue) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see section “Output in CSV format” on page 181).

Example 1

The FTAC administrator wishes to view the admission profile UMSAWARE with the command FTSHWPRF to determine if the profile might endanger data protection. He uses the following command.

```
FTSHWPRF˽NAME=UMSAWARE, -
      SELECT-PARAMETER=(OWNER-IDENTIFICATION=DONALD),INFORMATION=*ALL
```

Short form:

```
FTSHWPRF˽UMSAWARE,(,DONALD),*ALL
```

The output takes the following form:

```
UMSAWARE
  IGN-MAX-LEV = (IBR)
  FILE        = PROFIT
  USER-ADM    = (DONALD,M4711DON,OWN)
  PROC-ADM    = SAME
  SUCC-PROC   = NONE
  FAIL-PROC   = NONE
  DATA-ENC    = YES
  LAST-MODIF  = 2009-03-26 15:03:51
```

The first line shows the name of the admission profile. The next two lines show the settings which Donald made in the command FTCREPRF using the parameter IGNORE-MAX-LEVELS=(INBOUND-RECEIVE=*YES) and FILE-NAME= PROFIT. The values for USER-ADMISSION and PROCESSING-ADMISSION have not been set by Donald, but rather the default values have been used. The output SUCC-PROC=*NONE and FAIL-PROC=*NONE means that no follow-up processing is permitted. The output DATA-ENC=YES shows that Donald is especially careful, because this means that requests are only accepted if the user data is encrypted. Donald set this by using DATA-ENCRYPTION=*YES in the FTCREPRF command. The timestamp of the most recent change is shown under LAST-MODIF.

The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter FTMODPRF only with the parameter NAME, but no other parameters.

Example 2

The FTAC administrator examines the admission profile TESTPROF using the FTSHWPRF command to determine whether file processing is possible with this profile. The command is as follows:

```
FTSHWPRF␣NAME=TESTPROF, -
         SELECT-PARAMETER=(OWNER-IDENTIFICATION=DONALD),INFORMATION=*ALL
```

Short form:

```
FTSHWPRF␣TESTPROF,(,DONALD),INF=*ALL
```

The output has the following form:

```
TESTPROF
 INITIATOR   = REMOTE
 USER-ADM    = (DONALD,*FIRST,OWN)
 PROC-ADM    = SAME
 FT-FUNCTION = (TRANSFER-FILE,FILE-PROCESSING)
 LAST-MODIF  = 2009-01-31 15:03:44
```

The first line of the output displays the name of the admission profile. The second line indicates that the profile can only be used for requests initiated in the remote system. Donald has specified the value *FIRST for ACCOUNT in USER-ADMISSION; this means that the first account number assigned to the home pubset of the specified user ID in the system is used for account assignment in the case of transfer requests. As a result, it is unaffected by any changes to the account number. However, Donald has not specified a value for PROCESSING-ADMISSION and the default value SAME is therefore used. This means that the values are taken over from USER-ADMISSION. The FT-FUNCATION line indicates that the examined profile supports both pre-processing and file transfer requests. The last row specifies when the profile was last modified. The timestamp is also updated if you do not change the properties of the profile, i.e. if you enter FTMODPRF only with the parameter NAME, but no other parameters.

Example 3

The FT administrator wishes to view the profile REMADMIN that has been set up for remote administration by a remote administrator.

```
FTSHWPRF␣NAME=REMADMIN,INFORMATION=*ALL
```

Output has the following form:

```
REMADMIN
 USER-ADM    = (BS2ADMIN,,YES)
 FT-FUNCTION = (REMOTE-ADMINISTRATION)
 LAST-MODIF  = 2009-02-21 15:31:29
```

The output REMOTE-ADMINISTRATION for FT-FUNCTION indicates that the profile is permitted to perform remote administration. This means that the profile can be used by remote administrators to administer the local openFT instance. These remote administrators must also be configured in the remote administration server.

6.32 FTSHWPTN Display partner systems

Note on usage

User group: FT user and FT administrator

Functional description

The FTSHWPTN command is used to obtain the following information on partner systems included in the partner list of the current openFT instance:

– the names of the remote systems in the partner list,

– the status of the requests with the remote systems (activated or deactivated),

– priority assigned to the partner system,

– the setting for the openFT trace function on the partner system,

– the security level assigned to the remote system (this security level applies only if FTAC functionality is used. The information can then also be obtained using the FTSHWRGE command),

– the number of not yet completed file transfer requests submitted in the local system,

– the number of file transfer requests submitted in the remote systems for the local system,

– the partner address.

Format

```
FTSHWPTN

 PARTNER = *ALL / <text 1..200 with-low>
,OUTPUT = *STDERR(...) / *STDOUT(...)
    *STDERR(...) / *STDOUT(...)
      |  LAYOUT = *STD / *CSV / *BS2-PROC / *ZOS-PROC
,STATE = *ALL / *ACTIVE / *DEACT / *INSTALLATION-ERROR / *NO-CONNECTION / *NOT-ACTIVE /
             *AUTOMATIC-DEACTIVATION / *INACTIVE-BY-AUTOMATIC-DEACT
,INFORMATION = *STD / *ALL
```

Operands

PARTNER =
Partner system or systems about which information is to be output.

PARTNER = *ALL
Information on all partner systems is output.

PARTNER = <text 1..200 with-low>
Name or address of the partner system or group of partner systems about which information is to be output.
If you enter a name then you have two options:
You can either enter a unique partner name (1 - 8 alphanumeric characters) or a group of partners identified by a 1 to 7-character specification followed by an asterisk (*). For more information on partner addresses, see section “Defining partner properties” on page 108.

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or SYSERR, if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**omma **S**eparated **V**alue) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see page 181).

LAYOUT = *BS2-PROC
Output is supplied in the form of MODIFY-FT-PARTNER commands, which precisely define the partners involved. This enables the partner entries to be saved for a later reconstruction, to use them for an openFT operation on BS2000.

LAYOUT = *ZOS-PROC
Output is supplied in the form of FTMODPTN commands, which precisely define the partners involved. This enables the partner entries to be saved for a later reconstruction, to use them for an openFT operation on z/OS (see example on page 357).

STATE =
The scope of the output can be limited by the optional selection criteria in STATE. For an explanation of the selection criteria see page 355.

STATE = *ALL
The output is not limited by selection criteria.

STATE = *ACTIVE
All partner systems in the ACTIVE state are displayed.

STATE = *DEACT
All partner systems in the DEACT state are displayed.

STATE = *INSTALLATION-ERROR
All partner systems in the LUNK, RUNK, LAUTH, RAUTH, NOKEY and IDREJ state are displayed.

STATE = *NO-CONNECTION
All partner systems in the NOCON and DIERR state are displayed.

STATE = *NOT-ACTIVE
All partner systems not in the ACTIVE state are displayed.

STATE = *AUTOMATIC-DEACTIVATION
All partner systems are output which were assigned AUTOMATIC-DEACTIVATION.

STATE = *INACTIVE-BY-AUTOMATIC-DEACT
All partner systems are output which were actually deactivated using the option AUTOMATIC-DEACTIVATION.

INFORMATION = *STD / *ALL
Use this operand to control the scope of the information output. On *ALL, expanded address information is output, in addition to the standard information.

Example 1

Request information on all remote systems entered in the partner list:

```
FTSHWPTN INF=*STD
NAME     STATE SECLEV PRI  TRACE   LOC    REM P-CHK ADDRESS
HOSTABS2 ACT B-P-ATTR NORM FTOPT     0      0 FTOPT HOSTABS2
HOSTBBS2 ACT   STD    NORM FTOPT     0      0 FTOPT HOSTBBS2:102
PCUSER   ACT    40    LOW  FTOPT     0      0 FTOPT %IP123.23.99.120
PC1      ACT    40    LOW  FTOPT     0      0 FTOPT PC1
UNIX1    ACT    50    HIGH FTOPT     0      0 FTOPT UNIX1
UNIX2    ACT    50    HIGH FTOPT     0      0 FTOPT UNIX2:102
```

```
FTSHWPTN INF=*ALL
NAME     STATE SECLEV  PRI   TRACE   LOC    REM P-CHK ADDRESS        IDENTIFICATION
                                                      ROUTING
HOSTABS2 ACT B-P-ATTR  NORM  FTOPT     0      0 FTOPT HOSTABS2       HOSTABS2.FUSI.NET
HOSTBBS2 ACT   STD     NORM  FTOPT     0      0 FTOPT HOSTBBS2:102   HOSTBBS2.FUSI.NET
PCUSER   ACT    40     LOW   FTOPT     0      0 FTOPT %IP123.23.99.120
                                                                     %IP123.23.99.120
PC1      ACT    40     LOW   FTOPT     0      0 FTOPT PC1            PC1.FUSI.NET
UNIX1    ACT    50     HIGH  FTOPT     0      0 FTOPT UNIX1          UNIX1.FUSI.NET
UNIX2    ACT    50     HIGH  FTOPT     0      0 FTOPT UNIX2:102      %.UNIX2.$FJAM
```

The information displayed is explained below:

NAME
Symbolic names of the remote systems entered in the partner list.

STATE
Status of the partner system.

ACT
The partner system is active.

DEACT
The partner system is deactivated.

NOCON
The transport connection setup failed.

LUNK
The local system is unknown on the remote FT system.

RUNK
The partner system is unknown on the local transport system.

ADEAC
The partner system is active. It is deactivated if the connection cannot be established. This state is only displayed if STATE=*AUTOMATIC-DEACTIVATION has been specified; otherwise, these partner systems are maintained under the ACT status.

AINAC
The partner system was deactivated following several unsuccessful attempts to establish a connection. This status is only possible if STATE=*AUTOMATIC-DEACTIVATION has been specified.

LAUTH
The local system could not be authenticated in the partner system. A current, public key of the local openFT instance must be made available to the partner system.

RAUTH
The partner system could not be authenticated in the local system. A current, public key of the partner system must be imported to the SYSKEY library.

DIERR
A data integrity error was detected on the connection to the partner system. This can be due either to an error in the transport system, or to manipulation attempts along the transfer route. The connection was terminated but the affected request was not (if it is restartable).

NOKEY
The partner does not accept a connection without encryption, but no key is present in the local system. A new key must be created using FTCREKEY.

IDREJ
The partner or a go-between instance does not accept the instance ID sent from the local system. You must check to see if the local instance ID is consistent with the entry in the partner's partner list.

SECLEV
Security level assigned to the remote system when it was entered in the partner list. These security levels apply only if the FTAC-BS2000 is also implemented. STD stands for the default security level set with the FTMODOPT command. It applies to all partner systems which are entered in the partner list and for which nothing has been specifically defined.

PRI
Priority of a partner with respect to the processing of requests. The possible values are NORM, LOW and HIGH.

TRACE
Trace setting. You may specify the values ON, OFF and FTOPT (if FTMODPTN is specified, TRACE=*BY-FT-OPTIONS).

LOC
Number of FT requests that have been submitted in the local system and that address the FT system specified with REMSYS.

REM
Number of FT requests that have been submitted in the remote FT system and addressed to the local FT system. The remote system is specified in REMSYS.

IDENTIFICATION
Instance ID of the partner (also see the FTADDPTN command on page 195).

ROUTING
SESSION-ROUTING-INFO of the partner, where required (also see the FTADDPTN command, on page 195).

P-CHK
Type of sender checking for the current partner:

FTOPT
The global setting is valid.

T-A
The expanded sender checking is enabled for specific partners.

STD
The expanded sender checking is disabled for specific partners.

AUTH
With the aid of its public key in the SYSKEY library, the partner is subjected to an identity check ("authenticated") by cryptographic means.

NOKEY
Authentication is required and no key is present for the partner.

AUTHM
Authentication must be used.

ADDRESS
Partner address under which the remote system can be accessed. For more information on partner addresses, see section “Specifying partner addresses” on page 108.

Example 2

All partner entries in the partner list are to be saved in a form that will facilitate importing the entries into a different partner list as required. To do this, the output from the FTSHWPTN command is converted to the correct format using LAYOUT=*ZOS-PROC and piped to a file with the name PARTZOS.CLIST.

```
 READY
FREE DDNAME(SYSPRINT)
 READY
ALLOC DSNAME(PARTZOS.CLIST) DDNAME(SYSPRINT) NEW KEEP DSORG(PS) RECFM(F,B)
       LRECL(80)
 READY
FTSHWPTN OUTPUT=*STDOUT(LAYOUT=*ZOS-PROC)
 READY
FREE DDNAME(SYSPRINT)
```

If the partner systems are to be entered in a partner list again, this can be done using the TSO command EXEC:

```
EXEC PARTZOS
```

This method also provides a simple way of importing partner entries from a z/OS partner list into a BS2000 partner list. To do this, LAYOUT=*BS2-PROC must be specified in FTSHWPTN and the file that is generated must be transferred to BS2000 and executed there. In the same way, a file created in BS2000 (as of openFT V9.0) using SHOW-FT-PARTNER can be used to enter partner systems in the z/OS partner list.

6.33 FTSHWRGE Display partner systems

Note on usage

User group: FTAC user and FTAC administrator

Prerequisite for using this command is the use of openFT-AC.

Functional description

The command FTSHWRGE is used to list the partner systems with which you can communicate by file transfer. In addition to indicating the name of the partner system, the security level is output which the FT administrator assigned to this system in the partner list. To determine which basic functions you are permitted to use, you must use the command FTSHWADS to obtain information on your admission set (see page 304).

The FTAC administrator can use FTSHWRGE to list all partner systems with which his FT system can communicate using file transfer. Furthermore, he can find out for any user in his system which partner systems can be accessed by this user.

Format

```
FTSHWRGE

 USER-IDENTIFICATION = *OWN / <name 1..8>
,SELECT-PARAMETER = *ALL / *PARAMETERS(...)
   *PARAMETERS(...)
     |  PARTNER = *ALL / <text 1..200 with-low>
,OUTPUT = *STDERR(...) / *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
     |  LAYOUT = *STD / *CSV
```

Operands

USER-IDENTIFICATION =
User ID for which you would like to have a list of accessible partner systems.

USER-IDENTIFICATION = *OWN
The FTAC user receives all the partner systems with which he can use at least one basic function.
The FTAC administrator receives all accessible partner systems.

USER-IDENTIFICATION = <name 1..8>
The FTAC user can only enter his own user ID here, the output corresponds to *OWN.
The FTAC administrator can enter any user ID for which he would like to view the accessible partner systems.

SELECT-PARAMETER =
Specifies selection criteria for the partner systems.

SELECT-PARAMETER = *ALL
Obtains information on all partner systems which can be reached.

SELECT-PARAMETER = *PARAMETERS(PARTNER = <text 1..200 with-low>)
Obtains information on this partner system. You can specify the name from the partner list or the address of the partner system. Only two pieces of information are supplied:

- if you are permitted to communicate with this partner system.
- the security level assigned to this partner system.

For additional information to partner addresses, see section “Specifying partner addresses” on page 108.

OUTPUT =
Output medium for the partner system listing.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

> **LAYOUT = *STD**
> Output is put into a user-friendly form for reading.
>
> **LAYOUT = *CSV**
> Output is in **C**omma **S**eparated **V**alue format. This is a special tabular format, widely used in the PC world, where the individual fields are separated by a semicolon “;“ (see section “Output in CSV format” on page 181).

Example

Donald Duck would like to find out about the security level of the computer BUYDUCK. To do this, he uses the following command:

```
FTSHWRGE␣SELECT-PARAMETER=(PARTNER-NAME=BUYDUCK)
```

Short form:

```
FTSHWRGE␣SEL=(BUYDUCK)
```

He receives the following output:

```
SECLEV  PARTNER-NAME
  50    BUYDUCK
```

The column SECLEV contains the security level of the partner system whose name appears in the PARTNER-NAME column.
If Donald had entered SELECT-PARAMETER=*ALL (or left out this parameter altogether), he would have received a similar but longer list of all accessible partner systems.

6.34 FTSTART Activate openFT

Note on usage

User group: FT administrator

Functional description

The FTSTART command is used to activate the specified openFT instance once the openFT load module has been loaded and started. If the value "A" for automatic activation was specified in the openFT start parameters (see the description of the FJGEN command, page 182) then it is not necessary to enter the FTSTART command.

The command is only executed if openFT is not active.

If the request queue contains file transfer requests for which the corresponding (remote) FT systems have also been started, these requests are started directly after openFT starts – provided the resources are available and no other start time has been defined.

Adequate steps must also be taken to ensure that all file systems are available. Otherwise locally submitted requests that require unavailable file systems are terminated with an error message. If this happens, the user cannot be notified by a result list .

If the openFT instance is to run under a different host name, this host name must first be entered using FJGEN.

Format

FTSTART

Without operands

Correct execution of the FTSTART command is acknowledged with the following message:

```
FTR0500 OPENFT: openFT 11.0A00 starting. Protocols: openFT,FTP,ADM
```

6.35 FTSTOP Deactivate openFT

Note on usage

User group: FT administrator

Functional description

The FTSTOP is used to initiate deactivation of the specified openFT instance and stop openFT.

The command is only executed if the instance has been started.

Format

FTSTOP

Without operands

Correct execution of the FTSTOP command is acknowledged with the following message:

```
FTR0501 OPENFT: openFT terminated
```

Example

Activate the local openFT system and subsequently deactivate the FT system:

```
FTSTART
FTR0500 OPENFT: openFT V11.0A00 starting. Protocols: openFT, FTP, ADM
   .
   .
   .

FTSTOP
FTR0501 OPENFT: openFT terminated
```

6.36 FTTERM Terminate openFT

Note on usage

User group: FT administrator

FTTERM can be entered in the TSO command mode only.

Functional description

You use the FTTERM command to terminate the openFT load module.

Format

FTTERM

Without operands

FTTERM is acknowledged with the following message:

```
FTR4131 OPENFT: TERMINATION INITIATED BY USER
```
or
```
FTR4121 OPENFT: TERMINATED
```

Notes

- If openFT is still active at the time FJTERM is entered, it is deactivated before being terminated.
- Reactivation of the FT system after an FTTERM command is achieved by entering the commands FJINIT and FTSTART.

6.37 FTUPDKEY Update public keys

Note on usage

User group: FT administrator

Functional description

Using the FTUPDKEY command, you can newly create the public key files of the key pair sets present in your openFT instance. This may become necessary if the existing public key files are unintentionally deleted. In addition, the command imports updated comments from SYSPKF.COMMENT to the public key files (see below).

The key pair consists of a private key, which is administered internally by openFT, and a public key.

Public keys are stored under the name:

<openft qualifier>.<inst>.SYSPKF.R<key reference>.L<key length>.

Here, the first two name parts are replaced by OPENFT QUALIFIER and the name of the instance..

The key reference is a numeric designator for the version of the key pair. The key length is 768 by default. The public key files are text files that are created in the character code of the respective operating system, i.e. EBCDIC.DF04-1 for BS2000, IBM1047 for z/OS, ISO8859-1 for Unix systems and CP1252 for Windows systems.

In a file <openft qualifier>.<inst>.SYSPKF.COMMENT, you can store comments that are written in the first lines of this file when an existing public key file is updated. Such comments might contain, for example, the communications partner and the telephone number of the FT administrator on duty. The lines in the SYSPKF.COMMENT file may be a maximum of 78 characters in length.

Public key files with invalid key reference are automatically deleted (for example, public keys, for which openFT no longer has an internal private key).

Format

FTUPDKEY

Without operands

6.38 FTUPDPAR Update operating parameters

Note on usage

User group: FT administrator

Functional description

You can use this command to update certain settings in the parameter library while openFT is running. These are the specifications of IP addresses (TNSTCPIP member), the list of FT administrators (FTADM member) and the list of FTAC administrators (FTACADM member). Once you have edited these members you can take over your changes during operation with the FTUPDPAR command.

Format

FTUPDPAR

Without operands

6.39 NCANCEL Cancel file transfer requests

Note on usage

User group: FT user and FT administrator

Alias name: FTCANREQ

Functional description

The NCANCEL command can be used to cancel a file transfer request or to abort the file transfer. The FT system deletes from the request queue the file transfer request that corresponds to the specified selection criteria and, if necessary, aborts the associated file transfer.

The following features apply to this command:

– FT requests submitted either in the local or the remote system can be canceled.

– A single command can be used to cancel several FT requests simultaneously.

– The FT requests to be canceled can be selected using different selection criteria. When deleting a file transfer request only values actually specified in the FT request can be used as selection criteria.

– As FT administrator you can cancel requests from any user, whereas an FT user can only cancel those FT requests that he/she owns.

– As FT Administrator you can also fully and unconditionally cancel a selected request and remove it from the request file. “Unconditional” means that, if necessary, the request can be cancelled without any negotiation with the corresponding partner system. In this way, you can clear the request file of requests which are no longer recognized in the partner system or for which there is no longer any connection to the partner system.

WARNING!
If not used carefully, this function can result in inconsistencies in the request files of the corresponding partner systems. Under certain circumstances these inconsistencies may cause baffling error messages (SYSTEM ERROR) and “dead requests” in the partner system request files. It should therefore only be used in exceptional circumstances and after a suitable period has elapsed.

After the FT request is canceled, openFT initiates a follow-up processing in the event of failure (FAILURE-PROCESSING) which was previously specified in the NCOPY command. The following points apply:

- If you cancel a request issued in the local system, local FAILURE-PROCESSING will be initiated in any case; FAILURE-PROCESSING will be initiated in the remote system only if the data transfer process had already begun.
- If you cancel a request issued in a partner system, FAILURE-PROCESSING will be initiated both in the local and the remote system, respectively.

Note

- The user who issued the file transfer request in the local system is informed that the request has been aborted, provided that the FAILURE-PROCESSING operand was used in the original transfer request to specify user-generated result information for the local system, or if the default result list is to be supplied.
- The user in the remote system is only informed that the file transfer request has been aborted if file transfer has already been started and if the FAILURE-PROCESSING operand has been used to request user-generated result information for the remote system.
- The file transfer requests aborted with NCANCEL remain in the request queue until both systems involved have informed each other of the abort action.
- Requests for which the file transfer proper has already been completed but where the decision to end the request has not yet been reached with the partner can no longer be canceled.
- If a request is canceled while pre-processing or post-processing is running in z/OS, openFT starts a separate "Cancel-Job" to terminate the processing job. openFT constructs the cancel job with the TSOJOB job envelope from the openFT parameter library PARM. This job envelope is also required for follow-up processing (seepage 63) This Cancel-Job is assigned a "Z" as the last letter in the job name in order to give it a higher priority than the processing jobs that are currently running.

Format

```
NCANCEL / FTCANREQ

 TRANSFER-ID = *ALL / <integer 1..2147483639> (FORCE-CANCELLATION = *NO / *YES)
,SELECT = *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
   |  OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
   | ,INITIATOR = (*LOCAL, *REMOTE) / list-poss(2): *LOCAL / *REMOTE
   | ,PARTNER = *ALL / <text 1..200 with-low>
   | ,FILE-NAME = *ALL / <filename 1..59> / <c-string 1..512 with-low>
```

Meaning of operands

TRANSFER-ID =
Transfer ID of the FT request to be canceled.

TRANSFER-ID = *ALL
FT users can only delete FT requests of their own ID using this entry. FT administrators can delete all current FT requests that access the system.

TRANSFER-ID = <integer 1..2147483639>
Request identification which was communicated to the local system in the FT request confirmation. The associated FORCE-CANCELLATION parameter is available only to the FT administrator. It is used for an **unconditional** request cancellation.

TRANSFER-ID = <integer 1..2147483639>(FORCE-CANCELLATION = *NO)
NO is the default value. The request is removed from the request file following negotiation with the partner system.

TRANSFER-ID = <integer 1..2147483639>(FORCE-CANCELLATION = *YES)
The request is removed from the request file without negotiation with the partner system. For this to be possible, you must possess the FT-ADM privilege and the request must previously have been cancelled with CANCEL-FILE-TRANSFER FORCE=*NO.

SELECT =
Contains selection criteria for FT requests to be canceled. A request is canceled if it satisfies all the specified criteria.

SELECT = *OWN
Cancels all FT requests associated with the own user ID and the specified TRANSFER-ID.

SELECT = *PARAMETERS(...)

OWNER-IDENTIFICATION =
Designates the owner of the FT requests.

OWNER-IDENTIFICATION = *OWN
Cancels only the FT requests under the user's own ID.

OWNER-IDENTIFICATION = *ALL
Cancels FT requests under all user IDs. Only the administrator can use this entry.

OWNER-IDENTIFICATION = <name 1..8>
Specifies a particular user ID whose FT requests are to be canceled.

INITIATOR =
Initiator of the FT requests to be canceled.

INITIATOR = (*LOCAL,*REMOTE)
Cancels FT requests in the local system and in remote systems.

INITIATOR = *LOCAL
Cancels FT requests issued in the local system.

INITIATOR = *REMOTE
Cancels FT requests issued in remote systems.

PARTNER =
Cancels FT requests that were to be executed with a specific partner system.

PARTNER = *ALL
The name of the partner system is not used as a selection criterion to determine the FT requests to be canceled.

PARTNER = <text 1..200 with-low>
The FT requests that were to be executed with this partner are to be canceled.
The name must be specified in the same form in which it is output using NSTATUS.

FILE-NAME =
Cancels all FT requests in the local system that access this file or this library element whether as a send file or receive file. The file name or library member name must be specified exactly as it appears in the file transfer request.

FILE-NAME = *ALL
The file name is not used as a selection criterion to determine the FT requests to be canceled.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Cancels FT requests in the local system that access this file.

If multiple selection criteria are specified in the NCANCEL command, then each one of these must be valid for the requests that are to be canceled. Otherwise the NCANCEL command is acknowledged with the following message:

```
FTR0504 OPENFT: No requests available for the selection criteria.
```

Example 1

As an FT administrator, you want to cancel the request with the transfer ID 194578; you know that the user ID USER1 is the owner of this FT request. You issue the following command:

```
NCANCEL TRANSFER-ID=194578, SELECT=(OWNER=USER1)
```

A possible short form of this command is as follows:

```
NCAN 194578, (USER1)
```

openFT acknowledges the request with the following message:

```
FTR2072 OPENFT: Request 194578 has been canceled
```

Example 2

As an FT administrator, you want to cancel the request with the transfer ID 655423; you do not know who is the owner of this FT request. You issue the following command:

```
NCANCEL TRANSFER-ID=655423, SELECT=(OWNER=*ALL)
```

A possible short form of this command is as follows:

```
NCAN 655423, (*ALL)
```

Example 3

As an FT administrator, you want to cancel **all** FT requests involving your FT system. You issue the following command:

```
NCANCEL TRANSFER-ID=*ALL, SELECT=(OWNER=*ALL)
```

A possible short form of this command is as follows:

```
NCAN *ALL, (*ALL)
```

6.40 NSTATUS Query status of file transfer request

Note on usage

User group: FT user and FT administrator

Alias name: FTSHWREQ

Functional description

The NSTATUS command allows the FT user to request information about FT requests. As with NCANCEL, the user can specify selection criteria in order to obtain information about specific FT requests.

The FT administrator can obtain information about the requests of any owner. For this purpose, he must enter the NSTATUS command in FT administration mode.

The owner of requests issued in the local system is the user ID under which they are submitted. The owner of requests issued in the remote system is the user ID in the local system under which the requests are executed.

The scope of information to be output can be selected. By default the following information is output by the system in response to the NSTATUS command:

– the transfer ID of the request,
– the initiator of the request (local or remote system),
– the operating status of the request (see description of operands for more details),
– the partner system,
– the transfer direction,
– the name of the file to be transferred in the local system.

By entering INFORMATION=*ALL in the NSTATUS command more information can be obtained. openFT then, in addition to the standard output, gives the values of the following operands of the request (operands of the transfer command that was used to issue the request):

– `PRIO`
  With which priority is the request to be executed?

- `COMPRESS`
  Is the data to be transferred in compressed form?

- `WRITE`
  How is the receive file to be written?

- `START`
  What is the earliest start time for the request?

- CANCEL
  When is the request to be canceled?
- OWNER
  Who is the owner of the request?
- DATA
  How is the data to be transferred (binary/character)?
- TRANSP
  Should the data be transferred in transparent file format?
- ENCRYPT
  Should the user data be transferred in encrypted form?
- DICHECK
  Is data integrity checking in use?
- TARGFORM
  What file format is the file to be stored in on the target system?
- TRECFRM
  What record format is the file to be stored in on the target system?

- TRANSFER–ADMISSION (local)
  Which transfer admission for the local system was specified in the FT request?
- PROCESSING–ADMISSION (local)
  Which authorization for the follow-up processing in the local system was specified in the FT request?
- PROCESSING (local)
  Which commands were specified in the FT request for follow-up processing in the local system (SUCCESS/FAILURE)?

The following additional information is given for requests issued locally:

- LISTING
  Where is the result list to be sent?
- ASYN–MSG
  Specifies which request result leads to an asynchronous termination
  message.
  Possible values: ALL, FAIL.
  ALL is always specified here for file transfer requests issued from the command line interface. The openFT parameter library (ENDMSG_TO_TSO in PARM) controls whether the messages are actually issued.
- TRANSFER–ADMISSION (remote)
  Which transfer admission for the remote system was specified in the FT request?

- `PROCESSING-ADMISSION` (remote)
  Which authorization for follow-up processing in the remote system was specified in the FT request?
- `PROCESSING` (remote)
  Which commands were specified in the FT request for follow-up processing in the remote system (SUCCESS/FAILURE)?
- `CCSN` (local and remote)
  CCS name used in the local and/or remote system when reading the file.

See the examples at the end of this section for more details on how these operands appear in the command output.

The more precise your information request, the fewer irrelevant requests are output.

The specification INFORMATION=*SUMMARY returns a small table with the number of jobs in the various request states. See “Example 3” on page 382.

Format

```
NSTATUS / FTSHWREQ

 TRANSFER-ID = *ALL / <integer 1..2147483639>
,SELECT = *OWN / *PARAMETERS(...)
   *PARAMETERS(...)
   |   OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
   |  ,INITIATOR = (*LOCAL, *REMOTE) / list-poss(2): *LOCAL / *REMOTE
   |  ,PARTNER = *ALL(...) / <text 1..200 with-low>
   |     *ALL(...)
   |     |  PARTNER-STATE = *ALL / *ACTIVE
   |  ,FILE-NAME = *ALL / <filename 1..59> / <c-string 1..512 with-low>
   |  ,MONJV = *NONE /
   |  ,JV-PASSWORD = *NONE
   |  ,STATE = *ALL / *SUSPEND / *LOCKED / *WAIT / *ACTIVE / *FINISHED / *HOLD
,INFORMATION = *STD / *ALL / *SUMMARY
,OUTPUT = *STDERR(...) / *STDOUT(...)
   *STDERR(...) / *STDOUT(...)
   |   LAYOUT = *STD / *CSV
```

Operands

TRANSFER-ID =
Transfer ID of the FT request about which information is required.

TRANSFER-ID = *ALL
The FT administrator can obtain information about all current FT requests that access his system.

TRANSFER-ID = <integer 1..2147483639>
Transfer ID assigned to the local system and output as part of the message confirming acceptance of the request.

SELECT =
Contains selection criteria defining the file transfer requests on which inquiries are to be made. Information on a file transfer request is output if the request satisfies all the specified criteria.

SELECT = *OWN
Provides information on all current file transfer requests for which you are designated as the owner.

SELECT = *PARAMETERS(...)

> **OWNER-IDENTIFICATION =**
> Owner of the FT requests. Only the FT administrator can make use of this operand unrestricted.
>
> **OWNER-IDENTIFICATION = *OWN**
> Provides information only on the file transfer requests in the user’s own ID.
>
> **OWNER-IDENTIFICATION = *ALL**
> Provides information on FT requests in all user IDs.
>
> **OWNER-IDENTIFICATION = <name 1..8>**
> Specific user ID about whose file transfer requests information is required.
>
> **INITIATOR =**
> Initiator of the file transfer requests concerned.
>
> **INITIATOR = (*LOCAL,*REMOTE)**
> Provides information on file transfer requests in the local system and in remote systems.
>
> **INITIATOR = *LOCAL**
> Provides information on file transfer requests issued in the local system.
>
> **INITIATOR = *REMOTE**
> Provides information on file transfer requests issued in the remote systems.
>
> **PARTNER =**
> Selects file transfer requests carried out with a specified remote system.

PARTNER = *ALL(...)
The partner system is not used as a selection criterion to determine the file transfer requests on which information is to be output.

PARTNER-STATE =
The status of the partner system is used as a selection criterion.

PARTNER-STATE = *ALL
The requests are selected independently of the partner system's status.

PARTNER-STATE = *ACTIVE
Only the requests to and from the active partners are selected.

PARTNER = <text 1..200 with-low>
Name or an address of a partner system. Information is required on the file transfer requests being executed with this system. For more information on address specifications, see section "Specifying partner addresses" on page 108.

FILE-NAME =
FT requests that access this file in the local system as a send file or receive file. The file name or library member name must be specified exactly as it appears in the FT request. If %UNIQUE was specified, the file name generated by openFT must be entered as the selection criterion here.

FILE-NAME = *ALL
The file name is not used as a selection criterion to define the file transfer requests on which information is to be output.

FILE-NAME = <filename 1..59> / <c-string 1..512 with-low>
Name of a file. Information is required on the file transfer requests that access this file.

MONJV = *NONE
The parameter is supported for reasons of compatibility only.

JV-PASSWORD = *NONE
The parameter is supported for reasons of compatibility only.

STATE =
Selects those file transfer requests that are in the specified status. The status of a request may change in between entry of the command and information output. This is why the output may include requests that are in a state other than the one selected with STATE.

STATE = *ALL
The status of a request is not used as a selection criterion to define the file transfer requests on which information is to be output.

STATE = *SUSPEND
Requests information on those file transfer requests that are currently in SUSPEND status (= interrupted, e.g. by the command FTMODOPT STATE=*INACTIVE in the remote system or by a high-priority FT request).

STATE = *LOCKED
Requests information on FT requests that are in the LOCKED operating status (= temporarily locked as a result of a longer term resource bottleneck).

STATE = *WAIT
Requests information on those file transfer requests that are currently in WAIT status (= waiting for resources).

STATE = *ACTIVE
Requests information on those file transfer requests that are currently in ACTIVE status (= being processed).

STATE = *FINISHED
Requests information on those file transfer requests that are currently in FINISHED status (= terminated or aborted, but where the user has not yet been informed).

STATE = *HOLD
Requests information on those FT requests that are currently in HOLD status (= awaiting the specified start time).

INFORMATION =
Scope of the output.

INFORMATION = <u>*STD</u>
Output is summary form and contains the following information (see “Example 1” on page 377):

- Transfer ID,
- Status,
- Initiator,
- State,
- Partner,
- Direction of transfer,
- Byte count,
- File or library member name in the local system.

INFORMATION = *ALL
Output is in full form. In addition to the summary form data, further information is provided on the operands used in the NCOPY command (see “Example 2” on page 379).

INFORMATION = *SUMMARY
Output is in the form of a specified sum. By specifying INFORMATION=*SUMMARY, you can restrict the output information to a statistic of the currently existing requests. By doing this, the display is arranged according to the conditions in which the requests find themselves. The displayed sum can, of course, exceed the sum of the individual columns, since all requests, even those that still have no request condition, are counted.

Information is output about the number of request in each individual processing status (see “Example 3” on page 382).

OUTPUT =
Output medium.

OUTPUT = *STDERR(...)
Output is performed to SYSTSPRT or to SYSERR if this DDNAME is defined.

OUTPUT = *STDOUT(...)
Output is performed to SYSPRINT.

LAYOUT = *STD
Output is formatted using a standard layout that can be easily read by the user.

LAYOUT = *CSV
Output is supplied in CSV (**C**omma **S**eparated **V**alue) format. This is a widely used tabular format, especially in the PC environment, in which individual fields are separated by a delimiter, which is usually a semicolon “;“ (see section “Output in CSV format” on page 181).

If selection criteria are specified in the NSTATUS command and no request is found that matches all the specified criteria, the command is acknowledged with the following message:

```
FTR0504 OPENFT: No requests available for the selection criteria
```

Example 1

Information is to be output to SYSOUT on those FT requests submitted by the remote system ALFRED which require access to the file DRAISINE and are currently active. The required command is as follows:

```
NSTATUS SELECT=(INITIATOR=*REMOTE,PARTNER=ALFRED, -
FILE=DRAISINE,STATE=*ACTIVE)
```

The recommended short form of this command is as follows:

```
NSTATUS SEL=(INIT=*REM,PART-NAME=ALFRED,FILE=DRAISINE,STATE=*ACT)
```

The FT administrator must specify the operand OWNER=*ALL by SELECT, if he is not the owner of the file DRAISINE.

The information is then output in the following format, for example:

```
TRANS-ID   INI STATE PARTNER  DIR  BYTE-COUNT  FILE-NAME
528184     REM ACT   ALFRED   TO   14760       DRAISINE
```

Description of the output columns:

`TRANS-ID:` Transfer ID of the file transfer request

`INI:` Initiator of the file transfer request : *REM* for REMOTE, *LOC* for LOCAL

`STATE:` State of the request (here *ACT* for ACTIVE, other outputs:

`SUSP` for SUSPEND,
Inbound request suspended, e.g. due to higher priority requests.

`LOCK` for LOCKED,

`WAIT` for WAIT,

`FIN` for FINISHED,

`HOLD` for HOLD

`PARTNER:` Symbolic name of the relevant partner system.

If the FT request is in the STATE=WAIT state, and there is no normal internal resource bottleneck, then the partner name is preceded by one of the following characters:

* The FT administrator of the local system has locked a resource.

! An attempt to set up a connection to the partner system failed (possibly because the remote system is not running, for example, or because FT has not been started there or, in the case of TCP/IP connections, because the port specification contains *BY-TRANSPORTSYSTEM). This can also occur, if openFT has discovered an error during the internal check of transferred data integrity.

? Installation error.
Possible reasons:
- The remote system is connected to the local system via TCP/IP but the openFT connection to TCP/IP is interrupted.
- The authentication of the local or remote system has failed due to an unsuitable public key.

`DIR:` Transfer direction

`BYTE-COUNT:` Number of bytes transferred up to the last restart point (in the case of data compression this is the a number of bytes of compressed data)

`FILE-NAME:` Name of the relevant file or library member in the local system

Example 2

Full information is to be output to SYSPRINT on the FT request with transfer ID 524296. If the file transfer request was issued under the same user ID as that under which the inquiry is made, then the command is as follows:

```
NSTATUS TRANSFER-ID=524296,INFORMATION=*ALL,OUTPUT=*STDOUT
```

The recommended short form of this command is as follows:

```
NSTATUS TRANS=524296,INF=*ALL,OUT=*STDOUT
```

The information output on SYSLST then has the following format, for example:

```
TRANSFER-ID =524296      STORE  =09-04-22 11:18:05  FILESIZE=959
   STATE    =HOLD        BYTECNT=0
   INITIATOR=LOCAL       TRANS  =TO                 PRIO    =NORM
   WRITE    =REPLACE     START  =09-04-24 00:00:00  CANCEL  =NO
   COMPRESS =NONE        DATA   =CHAR
   TRANSP   =NO          ENCRYPT=NO
   TARGFORM =SEQ         TRECFRM=UNDEFINED
   OWNER    =SYSXYZ      DICHECK=NO
   PARTNER  =HUGO
   PARTNER-STATE =ACT
   PARTNER-PRIO  =NORM
   LOC: FILE     =BULLETIN
        TRANS-ADM=(USER,ACCOUNT)
        ASYN-MSG =ALL
        CCSN     =IBM1047
   REM: FILE     =TESTFILE
        TRANS-ADM=REMOTE-PROFILE
        CCSN     =EDF041
```

The fields in the output of the following meanings (by rows):

`TRANSFER-ID:` Transfer ID of the request

`STORE` The time at which the request was entered in the request queue

`FILESIZE:` The size of the file in bytes. If the output is flagged with a "K" on the right, the output is in kilobytes. If the output is flagged with "M", the output is in megabytes. The size is only shown here if the request has already been active. In the case of receive requests, a value is only shown here if the partner also sends that value.

`STATE:` State of the request

`BYTECNT:` Number of bytes transferred up to the last restart form (in the case of data compression in compressed form)

INITIATOR: Initiator of the request

TRANS: Transfer direction, as seen from local system

PRIO: Priority with which the request is to be started;
here: NORM for NORMAL.

WRITE: Specifies if or when the receive file is to be overwritten or extended

START: Requested start time of the request
(SOON for “as soon as possible”)

CANCEL: Requested abortion time
(NO for “no abortion requested”)

COMPRESS: Specifies whether or not the file is to be transferred in compressed form

DATA: Type of file:

- CHAR for text file
- BIN for binary file
- NOT-SPECIFIED
  in TRANSFER-FILE (NCOPY), no DATA-TYPE was specified
- USER for user format

TRANSP: Specifies whether the transfer is to be done in transparent format

ENCRYPT: Specifies whether the file content is to be transferred in encrypted form

TARGFORM: Format of the transferred file in the target system:

- SEQ Sequential file format
- BLOCK Block format

TRECFRM: Record format of the file in the target system:

- STD The same record format as in the sending system
- UNDEFINED
  Undefined record format

OWNER: Owner of request in local system

DICHECK: Specifies whether data integrity is to be checked (YES) or not (NO)

PARTNER: Symbolic name of partner system participating in the request.
If the FT request is in the STATE=WAIT state, and there is no normal internal resource bottleneck, then the partner name is preceded by one of the following characters:

* The FT administrator of the local system has locked a resource.

! An attempt to set up a connection to the partner system failed (possibly because the remote system is not running, for example, or because FT has not been started there or, in the case of TCP/IP connections, because the port specification contains *BY-TRANSPORTSYSTEM). This can also occur, if openFT has discovered an error during the internal check of transferred data integrity.

? Installation error.
Possible reasons:
- The partner system is connected to the local system via TCP/IP but the openFT connection to TCP/IP is interrupted.
- The authentication of the local or remote system has failed due to an unsuitable public key

PARTNER-STATE:
Status of the partner. Possible values:

ACT Activated

DEACT Deactivated

NOCON No connection, for instance because the openFT server has not been started on the remote system.

INSTERR
There is an installation or configuration error (for example, the local system is not known to the partner or the address of the partner in the partner list is not valid) or authentication of one of the partners has failed or encryption is not available locally or on the partner system.

PARTNER-PRIO:
Prioritization of the partner when processing requests.
Possible values:

LOW The partner has low priority.

NORM The partner has normal priority.

HIGH The partner has high priority.

LOC: Specifications on the local system (LOCAL-PARAMETER).

The entry can include more than in this example; the keywords correspond to the recommended abbreviations of the keywords of the transfer command; the meaning of the operand is also to be found there.

FILE: Local file name

ASYN-MSG:
Specifies which request result leads to an asynchronous termination message. Possible values: ALL, FAIL.

REM: Specifications on the remote system (REMOTE-PARAMETER).

The entry can include more than in this example; the keywords correspond to the recommended abbreviations of the keywords of the transfer command; the meaning of the operand is also to be found there.

FILE: Remote file name

The following parameters are only output for locally issued requests.

TRANS-ADM:
Transfer admission (here for the remote system. Instead of the triplet user ID, account number and password where appropriate, REMOTE-PROFILE can also be output here if a remote FTAC FT profile is addressed. The equivalent also applies to entries in the local system.

CCSN CCS name used in the local and/or remote system when reading the file.

Example 3

You want to output information about the number of request in each individual processing status.

```
NSTATUS INF=*SUMMARY
```

```
 ACT    WAIT    LOCK    SUSP   HOLD    FIN     TOTAL
   3      5       0       0      0      0         10
```

There are three requests in the ACTIVE condition, and five in the WAIT condition. Two requests are still in protocol handling, therefore the sum is 10.

Example 4

The FT administrator requires information about **all** FT requests affecting his FT system. If comprehensive information on FT requests is wanted, one possible short form of the command is:

```
NSTATUS SEL=(*ALL), INF=*ALL
```

If he only requires standard information about these FT requests, he may use one short form of the command as follows:

```
NSTATUS SEL=(*ALL)
```

The information is then output in the form (for example):

```
TRANS-ID   INI STATE PARTNER  DIR  BYTE-COUNT  FILE-NAME
242352178  LOC HOLD  TIGER    TO   0           'USER024.SRC'
242417736  REM ACT   JUMBO    FROM 128574      LISTING
242483296  REM SUSP  SYS435   TO   4582349     'USER832.FILE'
242548808  LOC ACT   XAS2     TO   765032      'PGM.LOAD'
242614296  LOC LOCK  TIGER    FROM 0           ASS.LIST
242679928  LOC WAIT  SYS435   TO   7776        'USER123.SRC'
242745512  LOC FIN   SIRIUS   TO   9457000     MONTH.STATS
```

If only the totals for all requests in the particular states is wanted, one possible short form of the command is:

```
NSTATUS SEL=(*ALL),INF=*SUM
```

The information is then output in the form (for example):

```
  ACT   WAIT   LOCK   SUSP   HOLD   FIN       TOTAL
    2      1      1      1      1     1           7
```

Example 5

The FT administrator has deactivated dynamic partners (DYN-PART=*OFF in the FTMODOPT command), but has entered a partner ANYBODY in the partner list to nevertheless allow access to partners that are not entered in the partner list.

Now information is required for all file transfer requests handled between the local system and anonymous partner systems. If the standard information scope is required, the following command must be entered:

```
NSTATUS SEL=(OWNER=*ALL,PARTNER=ANYBODY)
```

The FT administrator then receives the following output:

```
TRANS-ID   INI STATE PARTNER  DIR  BYTE-COUNT  FILE-NAME
557383872  REM ACT   ANYBODY  TO   14760       BRIEF.TXT
557449432  REM ACT   ANYBODY  FROM 88724       ASS.LST
```

7 Controlling via an operator console

openFT can be controlled from an operator console. For it to be possible to administer an openFT instance via the operator console, the ID *Console* must be entered in the FTADM and, if necessary, the FTACADM members of the PARM parameter library.

7.1 Starting openFT via an operator console

openFT can be started from an operator console in the usual way using the START command (started task):

```
START openft-procname
```

In this case, openft-procname is the name of the start procedure for the started task. An example of such a start procedure is given in the section “openFT as a job or started task” on page 85.

7.2 Terminating openFT via an operator console

openFT can be terminated from an operator console using the STOP command. The STOP command is converted internally into a FTSTOP and FTTERM command.

Command format:

```
STOP openft-jobname
```

`openft-jobname` name of the openFT batch job or started task.

openFT can also be canceled from the operator console in the normal way using the CANCEL command. openFT does not convert this command internally.

7.3 Issuing administration commands via an operator console

You can also enter administration commands at an operator console in order to control openFT. The commands are entered as follows:

```
MODIFY openft-jobname, adm-command
```

or in abbreviated form:

```
F openft-jobname, adm-command
```

`openft-jobname`: name of the openFT batch job or started task.

`adm-command`: FT administration command.

All the FT administration commands described in the chapter “Command interface for the FT administrator” on page 165 can be used here except FJGENPAR, FTHELP, FTSHWINS, FTSHWNET und FTTRACE.

If you enter the NCANCEL and NSTATUS commands at an operator console, they are interpreted as administration commands, i.e. you can use these commands to cancel and request information on all users' FT requests (privileged form of the command).

The FT administration commands must be entered as described in the chapter "Command interface for the FT administrator".

Only one MODIFY command can be processed at any one time. If another MODIFY command occurs during processing, the following message is issued:

```
MODIFY REJECTED-TASK BUSY
```

In this case you must repeat the command.

The messages issued by openFT in response to the administration commands are displayed at the operator console at which the command was entered. Message lines which do not begin with an error code (e.g. lines in the FTSHWPTN output) are prefixed with the code FJM2000. Since messages are output via the WTO macro in "single line" format, output consisting of a number of lines may be interspersed with other system messages.

Example

You want to set the two parameters CONNECTION-LIMIT and PROCESS-LIMIT to the value 2 from an operator console. Before you do so, you must deactivate the openFT system. After execution of the command, the FT system is to be reactivated. The name of the openFT batch job is USERAF.

You must enter the following commands at the operator console one after the other:

```
MODIFY USERAF, FTSTOP
MODIFY USERAF, FTMODOPT CONN-LIM=2, PROC-LIM=2
MODIFY USERAF, FTSTART
```

8 Controlling via NetView

openFT can be controlled via NetView or a NetView-compatible network management system (e.g. NetMaster). For it to be possible to administer an openFT instance via NetView, the ID *Console* must be entered in the FTADM and, if necessary, the FTACADM members of the PARM parameter library.

8.1 Starting openFT via NetView

openFT can be started under NetView as a started task. To this end, the system command START must be issued using the NetView command MVS:

```
MVS START openft-procname
```

In this case, openft-procname is the name of the start procedure for the started task. An example of such a start procedure is given in the section "openFT as a job or started task" on page 85.

8.2 Terminating openFT via NetView

You can also terminate openFT under NetView by issuing the STOP command as a system command. The STOP command is converted internally into an FTSTOP and FTTERM command.

Command format:

```
MVS STOP openft-jobname
```

`openft-jobname` name of the openFT batch job or started task.

In addition, the CANCEL command can be issued via NetView as a system command, thus canceling openFT. openFT does not convert this command internally.

8.3 Issuing administration commands via NetView

The MODIFY command can also be issued via NetView as a system command. Administration commands for controlling openFT can thus be entered as follows:

```
MVS MODIFY openft-jobname, adm-command
```

or in abbreviated form: `MVS F openft-jobname, adm-command`

`openft-jobname`:	name of the openFT batch job or started task
`adm-command`:	FT administration command

All the FT administration commands described in the chapter “Command interface for the FT administrator” on page 165 can be used here except FJGENPAR, FTHELP, FTSHWINS, FTSHWNET und FTTRACE.

If you enter the NCANCEL and NSTATUS commands at an operator console, they are interpreted as administration commands, i.e. you can use these commands to cancel and request information on all users' FT requests (privileged form of the command).

The FT administration commands must be entered as described in the chapter "Command interface for the FT administrator".

Only one MODIFY command can be processed at any one time. If another MODIFY command occurs during processing, the following message is issued:

```
MODIFY REJECTED-TASK BUSY
```

In this case you must repeat the command.

The messages issued by openFT in response to the administration commands are sent to the NetView console at which the command was entered. Message lines which do not begin with an error code (e.g. lines in the FTSHWPTN output) are prefixed with the code FJM2000. The messages can then be processed using NetView-specific functions. Since messages are output via the WTO macro in "single line" format, output consisting of a number of lines may be interspersed with other system messages.

Example

You want to set the two parameters CONNECTION-LIMIT and PROCESS-LIMIT on the value 2 under NetView. In addition, the partner HOSTA is to be deactivated. The name of the openFT batch job is USERAF. You must enter the following commands at the NetView console one after the other:

```
MVS MODIFY USERAF, FTMODOPT CONN-LIM=2, PROC-LIM=2
MVS MODIFY USERAF, FTMODPTN HOSTA,STATE=*DEACT
```

9 Appendix

9.1 Comparison of the old and new administration commands

The "old" FT administration interface via the FJADM program and its commands is no longer supported in the current version It is replaced by the new TSO commands. Commands that are absent from the following list will continue to be supported because they are not FJADM statements.

Old command	New command	Function
END	withdrawn	End FT administration mode
FJACTSYS	FTMODPTN STATE=	Activate requests to a remote system
FJADDSYS	FTADDPTN	Enter remote system in the partner list
FJADM	withdrawn	Activate administration mode
FJCREATEFSF	FTCRERQF	The function has been withdrawn because the request queue and the partner list (was previously the network description file) are set up on installation.
FJDEACTSYS	FTMODPTN STATE=	Deactivate requests to a remote system
FJDELETEFSF	withdrawn	The function has been withdrawn because the request queue and the partner list (was previously the network description file) can no longer be deleted.
FJMODPAR	FTMODOPT	Modify operating parameters
FJPARINFO	FTSHWOPT	Obtain information about operating parameters
FJREMOVESYS	FTREMPTN	Remove remote system from the partner list
FJRUN	withdrawn	Execute sequence of FT administration commands
FJSHUTDOWN	FTSTOP	Deactivate local FT system
FJSNAP	is omitted in future	Initiate openFT dump
FJSTART	FTSTART	Activate local FT system
FJSYSINFO	FTSHWPTN	Obtain information about remote systems

Old command	New command	Function
FJTERM	FTTERM	End openFT
FJTRACEOFF	FTMODOPT TRACE=*OFF	Deactivate trace function
FJTRACEON	FTMODOPT TRACE=*ON(...)	Activate trace function

9.2 Structure of CSV outputs

9.2.1 Output format

The output format for all commands corresponds to the following rules:

- Each record is output in a separate line. A record contains all the information to be displayed on an object.
- The first line is a header and contains the field names of the respective columns. **Only the field names are guaranteed, not the order of fields in the record.** In other words, the order of columns is determined by the order of the field names in the header line.
- The FTSHWENV command outputs two tables one after the other each with their own header. If a table is empty then the corresponding header is omitted.
- Individual fields within an output line are delimited by a semicolon ";".

The following data types are differentiated in the output:

- Numeric
- String:
  Since the ";" character in the CSV output is a metacharacter, text containing a ";" is enclosed within double quotes ("). Double quotes appearing within a text field are duplicated in order to differentiate them from text delimiters. When the text is imported into a program, all such duplicated quotes are eliminated, and all text delimiters are removed. Keywords are output in uppercase, starting with an asterisk (*), and are not enclosed within double quotes.
- Date:
  The date and time are output in the form yyyy-mm-dd hh:mm:ss or only as yyyy-mm-dd.

The tables in the following sections show the fields that appear in the CSV output of the individual commands. The scope of the output can be specified for the commands FTSHWLOG and FTSHWPRF. In this case, the fields that appear independent of the selected output scope are shown first. These are followed by the fields that only appear when the extensive output version is chosen. The sequence in which the fields appear in the table therefore does not correspond to the sequence in which they are displayed by the command.

Some of the fields in this command output are irrelevant for openFT for z/OS, but they appear nonetheless for reasons of compatibility with other openFT products (e.g. ElemName, ElemPrefix etc0. in the output of FTSHWPRF).

9.2.2 FTSHWADS

Column	Type	Values
UserId	String	Value
UserMaxObs	Numeric	Value
UserMaxObsStd	String	*YES / *NO
UserMaxObr	Numeric	Value
UserMaxObrStd	String	*YES / *NO
UserMaxIbs	Numeric	Value
UserMaxIbsStd	String	*YES / *NO
UserMaxIbr	Numeric	Value
UserMaxIbrStd	String	*YES / *NO
UserMaxIbp	Numeric	Value
UserMaxIbpStd	String	*YES / *NO
UserMaxIbf	Numeric	Value
UserMaxIbfStd	String	*YES / *NO
AdmMaxObs	Numeric	Value
AdmMaxObsStd	String	*YES / *NO
AdmMaxObr	Numeric	Value
AdmMaxObrStd	String	*YES / *NO
AdmMaxIbs	Numeric	Value
AdmMaxIbsStd	String	*YES / *NO
AdmMaxIbr	Numeric	Value
AdmMaxIbrStd	String	*YES / *NO
AdmMaxIbp	Numeric	Value
AdmMaxIbpStd	String	*YES / *NO
AdmMaxIbf	Numeric	Value
AdmMaxIbfStd	String	*YES / *NO
Priv	String	*YES / *NO
Password	String	*YES / *NO
AdmPriv	String	*YES / *NO

9.2.3 FTSHWENV

The FTSHWENV command outputs the objects present in an FTAC export file in a format that corresponds to the output from the FTSHWADS (page 304) and FTSHWPRF (page 402) commands.

9.2.4 FTSHWLOG

Column	Type	Values	INF =
LogId	Numeric	Value	*STD / *ALL
ReasonCode	String	Value enclosed within double quotes to prevent interpretation as a number. FTAC Reason Codes are output in the form of a Hex string	
LogTime	yyyy-mm-dd hh:mm:ss	Value	
InitUserId	String	Value / *REM	
InitTsn	String	Value / *NONE	
PartnerName	String	Value	
TransDir	String	*TO / *FROM / *NSPEC	
RecType	String	*FT / *FTAC / *ADM	
Func	String	*TRANSFILE / *READ-FILE-ATTR / *DEL-FILE / *CRE-FILE / *MOD-FILE-ATTR / *READ-DIR / *MOVE-FILE / *CRE-FILE-DIR / *DEL-FILE-DIR / *MOD-FILE-DIR / *LOGIN / *REM-ADMIN	
UserAdmisId	String	Value	
FileName	String	Value	

Column	Type	Values	INF =
Priv	String	*NO / *YES for FTAC logging records and with entry of an FTAC profile; otherwise, *NONE	*ALL
ProfName	String	Value / *NONE	
ResultProcess	String	*NONE / *STARTED / *NOT-STARTED	
StartTime	yyyy-mm-dd hh:mm:ss	Value	
	String	*NONE	
TransId	Numeric	Value	
	String	*NONE	
Write	String	*REPL / *EXT / *NEW / *NONE	
StoreTime	yyyy-mm-dd hh:mm:ss	Value	
	String	*NONE	
ByteNum	Numeric	Value	
	String	*NONE	
DiagInf	String	*NONE / Value	
ErrInfo	String	Value / empty	
Protection	String	*STD / *SAME	
SecEncr	String	*YES / *NO	
SecDichk	String	*YES / *NO	
SecDencr	String	*YES / *NO	
SecDdichk	String	*YES / *NO	
SecLauth	String	*YES / *NO	
SecRauth	String	*YES / *NO	
RsaKeyLen	Numeric	Value	
SymEncrAlg	String	*DES / *AES-128 / *AES-256	
CCSName	String	Value / empty	
AdminId	String	Value	
Routing	String	Value	
AdmCmd	String	Value	

9.2.5 FTSHWMON

The following table lists the monitoring values displayed with INFORMATION=*VALUES(...). For a detailed description of the monitoring values, refer to the section "Description of the monitoring values" on page 330.

If DATA=*RAW is specified, the duration values are not output (*Duxxx*, see footnote).

Column	Type [1]	Value [1]	INF=
CurrTime	yyyy-mm-dd hh:mm:ss	current time	*STD/ *ALL
MonOn	yyyy-mm-dd hh:mm:ss	start time of monitoring	
PartnerSel	String	*ALL / OPENFT / FTP / *NONE	
ReqSel	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE	
Data	String	FORM / RAW	
ThNetbTtl	Numeric	Value	
ThNetbSnd	Numeric	Value	
ThNetbRcv	Numeric	Value	
ThNetbTxt	Numeric	Value	*ALL
ThNetbBin	Numeric	Value	
ThDiskTtl	Numeric	Value	*STD/ *ALL
ThDiskSnd	Numeric	Value	
ThDiskRcv	Numeric	Value	
ThDiskTxt	Numeric	Value	*ALL
ThDiskBin	Numeric	Value	
ThRqto	Numeric	Value	*STD/ *ALL
ThRqft	Numeric	Value	*ALL
ThRqfm	Numeric	Value	
ThSuct	Numeric	Value	*STD/ *ALL
ThAbrt	Numeric	Value	
ThIntr	Numeric	Value	
ThUsrf	Numeric	Value	

Column	**Type** [1]	**Value** [1]	**INF=**
ThFoll	Numeric	Value	*ALL
ThCosu	Numeric	Value	
ThCofl	Numeric	Value	*STD/ *ALL
ThCobr	Numeric	Value	
DuRqtlOut[2]	Numeric	Value	*ALL
DuRqtlInb[2]	Numeric	Value	
DuRqftOut[2]	Numeric	Value	
DuRqftInb[2]	Numeric	Value	
DuRqfmOut[2]	Numeric	Value	
DuRqfmInb[2]	Numeric	Value	
DuRqesOut[2]	Numeric	Value	
DuDnscOut[2]	Numeric	Value	
DuDnscInb[2]	Numeric	Value	
DuConnOut[2]	Numeric	Value	
DuOpenOut[2]	Numeric	Value	
DuOpenInb[2]	Numeric	Value	
DuClosOut[2]	Numeric	Value	
DuClosInb[2]	Numeric	Value	
DuUsrcOut[2]	Numeric	Value	
DuUsrcInb[2]	Numeric	Value	
StRqas	Numeric (100)[3]	Value	*STD/ *ALL
StRqaa	Numeric (100)[3]	Value	
StRqwt	Numeric (100)[3]	Value	
StRqhd	Numeric (100)[3]	Value	
StRqsp	Numeric (100)[3]	Value	
StRqlk	Numeric (100)[3]	Value	
StRqfi	Numeric (100)[3]	Value	*ALL

Column	Type [1]	Value [1]	INF=
StCLim	Numeric	Value	*STD/ *ALL
StCAct	Percent	Value	
StRqLim	Numeric	Value	
StRqAct	Percent	Value	
StOftr	BOOL	1 (openFT protocol active) / 0 (not active)	
StFtmr	BOOL	1 (FTAM protocol active) / 0 (not active)	
StFtpr	BOOL	1 (FTP protocol active) / 0 (not active)	
StTrcr	BOOL	1 (Trace switched on) / 0 (switched off)	*ALL

[1] Applies to formatted output. In the case of raw output, the representation or the way in which values are derived sometimes differs.

[2] This value is not output with raw output.

[3] Numeric(100) has the following meaning in CSV format: The monitoring value is multiplied by 100 and output (thus, for example, the output value 225 corresponds to the monitoring value 2.25). In table format, the value is output with two decimal places.

9.2.6 FTSHWOPT

Column	**Type**	**Values**
PartnerLim	Numeric	Value
ReqLim	Numeric	Value
TaskLim	Numeric	Value
ConnLim	Numeric	Value
ReqWaitLev	Numeric	Value
TransportUnitSize	Numeric	Value
PartnerCheck	String	*STD / *TRANSP-ADDR
SecLev	Numeric	Value
	String	*B-P-ATTR
TraceOpenft	String	*STD / *OFF
TraceOut [1]	String	*FILE / empty
TraceSession	String	*OFF
TraceFtam	String	*OFF / *STD
LogTransFile	String	*OFF / *ON
MaxInboundReq	Numeric	Value
MaxReqLifetime	String	*UNLIMITED / Value
SnmpTrapsSubsystemState	String	*OFF / *ON
SnmpTrapsFtState	String	*OFF / *ON
SnmpTrapsPartnerState	String	*OFF / *ON
SnmpTrapsPartnerUnreach	String	*OFF / *ON
SnmpTrapsReqQueueState	String	*OFF / *ON
SnmpTrapsTransSucc	String	*OFF / *ON
SnmpTrapsTransFail	String	*OFF / *ON
ConsoleTraps	String	*OFF / *ON
TeleService	String	FT / FTAC / FT/FTAC
HostName	String	Value
Identification	String	Value
UseTns	String	*YES / *NO
ConsTrapsSubsystemState	String	*ON / *OFF
ConsTrapsFtState	String	*ON / *OFF
ConsTrapsPartnerState	String	*ON / *OFF
ConsTrapsPartnerUnreach	String	*ON / *OFF

Column	Type	Values
ConsTrapsReqQueueState	String	*ON / *OFF
ConsTrapsTransSucc	String	*ON / *OFF
ConsTrapsTransFail	String	*ON / *OFF
FtLog	String	*ALL / *NONE / *FAIL
FtacLog	String	*ALL / *FAIL / *MODIFY
Trace	String	*ON / *OFF
TraceSelp	String	*ALL / *NONE / OPENFT / FTP / ADM [2]
TraceSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [2]
TraceOpt	String	*NONE / Value
KeyLen	Numeric	Value
CcsName	String	Value
AppEntTitle	String	*YES / *NO
StatName	String	Value
SysName	String	Value / empty
FtStarted	String	*YES / *NO
openftAppl	String	*STD / Value
ftamAppl	String	*STD / Value
FtpPort	Numeric	Value / empty
ftpDPort	Numeric	Value / empty (internal function)
ftstdPort	Numeric	Value
DynPartner	String	*ON / *OFF
ConTimeout	Numeric	Value (internal function)
ChkpTime	Numeric	Value (internal function)
Monitoring	String	*ON / *OFF
MonSelp	String	*ALL / OPENFT / FTP / empty [2]
MonSelr	String	*ALL / ONLY-SYNC / ONLY-ASYNC / ONLY-LOCAL / ONLY-REMOTE [2]
AdmTrapServer	String	Value / *NONE
AdmTrapsFtState	String	*ON / *OFF
AdmTrapsPartnerState	String	*ON / *OFF
AdmTrapsPartnerUnreach	String	*ON / *OFF
AdmTrapsReqQueueState	String	*ON / *OFF
AdmTrapsTransSucc	String	*ON / *OFF

Column	Type	Values
AdmTrapsTransFail	String	*ON / *OFF
AdminConnLim	String	Value
AdmPort	String	Value / *NONE
OpenftApplState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL
FtamApplState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL
FtpState	String	*ACTIVE / *INACT / *DISABLED / *NAVAIL
AdmState	String	*ACTIVE / *INACT / *DISABLED
AdminLog	String	*ALL / *FAIL / *MODIFY / *NONE
CentralAdminServer	String	*YES / *NO [3]
ActiveAppl	String	*ALL / *NONE / OPENFT / FTP / ADM [2]

[1] TraceOut only applies with openFT partners

[2] Comma-separated combinations of multiple values are also possible (not with *ALL or *NONE).

[3] Always *NO for openFT for z/OS

9.2.7 FTSHWPRF

Column	Type	Values	INF=
ProfName	String	Value	*ONLY-NAMES / *ALL
Priv	String	*YES / *NO	
TransAdm	String	*NSPEC / *SECRET	
Duplicated	String	*YES / *NO	
LockedByImport	String	*YES / *NO	
LockedByAdm	String	*YES / *NO	
LockedByUser	String	*YES / *NO	
Expired	String	*YES / *NO	
UserAdmId	String	Value	
UserAdmAcc	String	Value / *FIRST/*NSPEC / *NRES	*ALL
UserAdmPass	String	*OWN / *NSPEC / *NONE / *YES	
ExpDate	yyyy-mm-dd	Value / *NRES	
Usage	String	*PUBLIC / *PRIVATE / *NSPEC	
IgnObs	String	*YES / *NO	
IgnObr	String	*YES / *NO	
IgnIbs	String	*YES / *NO	
IgnIbr	String	*YES / *NO	
IgnIbp	String	*YES / *NO	
IgnIbf	String	*YES / *NO	
Initiator	String	*LOC / *REM / *NRES	
TransDir	String	*FROM / *TO / *NRES	
MaxPartLev	Numeric	Value	
	String	*NRES	
Partners	String	One or more FT partners, separated by commas / *NRES	
FileName	String	Value / *NRES	
Library	String	*YES / *NO / *NRES / Value	
FileNamePrefix	String	*YES / *NO	
ElemName	String	Value / *NRES / *NONE	
ElemPrefix	String	*YES / *NO	
ElemVersion	String	Value / *STD / *NONE	
ElemType	String	Value / *NRES / *NONE	

Column	Type	Values	INF=
FilePass	String	*YES / *NRES / *NONE	*ALL
Write	String	*NEW / *EXT / *REPL / *NRES	
ProcAdmId	String	Value / *NRES / *SAME	
ProcAdmAcc	String	Value / *NRES / *SAME	
ProcAdmPass	String	*NONE / *YES / *NRES / *SAME	
SuccProc	String	Value / *NONE / *NRES / *EXPANSION	
SuccPrefix	String	Value / *NONE	
SuccSuffix	String	Value / *NONE	
FailProc	String	Value / *NONE / *NRES /*EXPANSION	
FailPrefix	String	Value / *NONE	
FailSuffix	String	Value / *NONE	
TransFile	String	*ALLOWED / *NOT-ALLOWED	
ModFileAttr	String	*ALLOWED / *NOT-ALLOWED	
ReadDir	String	*ALLOWED / *NOT-ALLOWED	
FileProc	String	*ALLOWED / *NOT-ALLOWED	
AccAdm	String	*ALLOWED / *NOT-ALLOWED	
RemAdm	String	*ALLOWED / *NOT-ALLOWED	
Text	String	Value / *NONE	
DataEnc	String	*YES / *NO / *NRES	
ModDate	yyyy-mm-dd hh:mm:ss	Value	
AdmTrapLog	String	*ALLOWED / *NOT-ALLOWED	

9.2.8 FTSHWPTN

Column	Type	Values
PartnerName	String	Value
Sta	String	*ACT / *DEACT / *NOCON / *LUNK / *RUNK / *ADEAC / *AINACT / *LAUTH / *RAUTH / *NOKEY / *DIERR / *IDREJ
SecLev	String	*STD / *B-P-ATTR
	Numeric	Value
Trace	String	*FTOPT / *ON / *OFF
Loc	Numeric	Value
Rem	Numeric	Value
Processor	String	Value
Entity	String	Value
NetworkAddr	String	Value
Port	Integer	Value
PartnerCheck	String	*FTOPT / *STD / *TRANSP-ADDR / *AUTH / *AUTHM / *NOKEY
TransportSel	String	Value
LastAccessDate	yyyy-mm-dd	Value
NetworkAddr	String	Value
TransportSel	String	Value
SessionSel	String	Value / *NONE
PresentationSel	String	Value / *NONE
Identification	String	Value
SessRout	String	Value / *ID / empty
PartnerAddr	String	Value
Check	String	*FTOPT / *STD / *TRANSP-ADDR
AuthMand	String	*YES / *NO
Priority	String	*LOW / *NORM / *HIGH

9.2.9 FTSHWRGE

Column	Type	Values
SecLev	Numeric	Security level setting
PartnerName	String	Partner name

9.2.10 NSTATUS

Column	Type	Values	INF=
TransId	Numeric	Value	*STD/ *ALL
Initiator	String	*LOC / *REM	
State	String	*SUSP / *LOCK / *WAIT / *HOLD / *FIN / *ACT	
PartnerName	String	Value	
PartnerState	String	*ACT / *INACT / *NOCON / *INSTERR	
TransDir	String	*TO / *FROM	
ByteNum	Numeric	Value	
LocFileName	String	*NSPEC / Value	
LocElemName	String	*NSPEC / *NONE / Value	
LocElemType	String	*NSPEC / *NONE / Value	
LocElemVersion	String	*STD / *NONE / Value	
Prio	String	*LOW / *NORM / *HIGH	*ALL
Compress	String	*NONE / *BYTE / *ZIP	
DataEnc	String	*YES / *NO	
DiCheck	String	*YES / *NO	
Write	String	*REPL / *EXT / *NEW	
StartTime	yyyy-mm-dd hh:mm:ss	Value	
	String	*SOON / Value	
CancelTime	yyyy-mm-dd hh:mm:ss	Value	
	String	*NO	
Owner	String	Value	
DataType	String	*CHAR / *BIN / *NSPEC	
Transp	String	*YES / *NO	

Column	Type	Values	INF=
LocTransAdmId	String	*NONE / Value	*ALL
LocTransAdmAcc	String	*NONE / Value	
LocProfile	String	*NONE / Value	
LocProcAdmId	String	*NONE / Value	
LocProcAdmAcc	String	*NONE / Value	
LocSuccProc	String	*SECRET / *NONE / Value	
LocFailProc	String	*SECRET / *NONE / Value	
LocListing	String	*NONE / *SYSLST / *LISTFILE	
LocMonjv	String	*NONE / Value	
LocCcsn	String	*STD / Value	
RemFileName	String	*NSPEC / *NONE / Value	
RemElemName	String	*NSPEC / *NONE / Value	
RemElemType	String	*NSPEC / *NONE / Value	
RemElemVersion	String	*STD / *NONE / Value	
RemTransAdmId	String	*NONE / Value	
RemTransAdmAcc	String	Value / empty	
RemTransAdmAccount	String	Value / empty	
RemProfile	String	*YES / *NONE	
RemProcAdmId	String	*NONE / Value	
RemProcAdmAcc	String	*NONE / Value	
RemSuccProc	String	*SECRET / *NONE / Value	
RemFailProc	String	*SECRET / *NONE / Value	
RemCcsn	String	*STD / Value	
FileSize	Numeric	Value / empty	
RecSize	Numeric	Value / empty	
RecFormat	String	*STD / *VARIABLE / *FIX / *UNDEFINED	
StoreTime	yyyy-mm-dd hh:mm:ss	Value	
ExpEndTime	yyyy-mm-dd hh:mm:ss	Value / empty	
TranspMode	String	*YES / *NO	
DataEncrypt	String	*YES / *NO	
TabExp	String	*AUTO / *YES / *NO	
Mail	String	*ALL / *FAIL / *NO	

Column	Type	Values	INF=
DiagCode	String	Value / empty	*ALL
FileAvail	String	*IMMEDIATE / *DEFERRED / *NSPEC	
StorageAccount	String	Value / empty	
AccessRights	String	Value / empty	
LegalQualif	String	Value / empty	
PartnerPrio	String	*LOW / *NORM / *HIGH	
TargetFileForm	String	*STD / *BLOCK / *SEQ	
TargetRecForm	String	*STD / *UNDEFINED	
Protection	String	*STD / *SAME	

Column	Type	Values	INF=
Act	Numeric	Value	*SUMMARY
Wait	Numeric	Value	
Lock	Numeric	Value	
Susp	Numeric	Value	
Hold	Numeric	Value	
Fin	Numeric	Value	
Total	Numeric	Value	

9.3 Accounting records

Structure of openFT **accounting records**

An openFT accounting record is divided into the following parts:

– SMF header
– record definition
– product information
– FT administrator area
– user information
– basic information
– file information

The following description of these record sections includes the absolute and relative **offsets** (relative to the start of the SMF record or the start of the record section being described), the **length** (in bytes) and the **format** of the data field. The following abbreviations are used when specifying the formats:

A alphanumeric

B binary

C printable character

F file name for z/OS

P packed decimal number

Z unpacked decimal number

Layout of the SMF header

Offsets (hex.)		**Length** (dec.)	**Format**	**Description**
abs	rel			
00	00	2	B	Length of the SMF record (including the length field) (1)
02	02	2	B	Segment descriptor (1)
04	04	1	B	System indicator "0xxxxx10" = OS/VS2 (2)
05	05	1	B	Record type (128, ..., 255)
06	06	4	B	Record storage time in hundredths of seconds since 0:00 local time
0A	0A	4	P	Record storage date in the format 0CYYDDDF (3)
0E	0E	4	C	System ID (from the SID parameter)

(1) The fields "length of the SMF record" and "segment descriptor" together form the record descriptor word (RDW). Depending on the reading method used, the RDW may be missing from the SMF records read out. The segment descriptor is set to "0000", i.e. only non-spanned records are written.

(2) Bits specified with "x" are reserved by IBM and are set by SMF under certain circumstances.

(3)

C :	centuries later than the 20th century
YY :	year
DDD:	days in year
F :	sign (= X'F')

Layout of the record definition section

Offsets (hex.)		**Length** (dec.)	**Format**	**Description**
abs	rel			
12	00	4	A	Record ID ("FTR0")
16	04	2	C	Record version ("1A") (4)
18	06	2	B	Offset for product information (5)
1A	08	2	B	Offset for FT administrator area (5)
1C	0A	2	B	Offset for user information (5)
1E	0C	2	B	Offset for basic information (5)
20	0E	2	B	Offset for file information (5)

(4) An analysis program can recognize the structure of the accounting record from the record version. The structure described here corresponds to version "1A"; accounting records with a different structure (subsequent versions) are identified where appropriate by the corresponding record versions. The following is guaranteed for record versions "1A", "1B" etc.:

- The order in which the offset information is described here is retained in the record definition section.
- The structure of the record sections described here (product information, FT administrator area, etc.) is retained; if necessary, additional information is appended at the end of the relevant record section.

(5) These offsets are given in relation to the start of the SMF record. If, after the SMF record has been read, the record descriptor word (see above) is missing, 4 bytes must be subtracted from the specified offsets.

Layout of the product information

Offsets (hex.)		**Length** (dec.)	**Format**	**Description**
abs	rel			
22	00	6	C	Product name ("openFT")
28	06	4	C	Product version ("90A0")

Layout of the FT administrator area

Offsets (hex.)		**Length** (dec.)	**Format**	**Description**
abs	rel			
2C	00	40	C	FT administrator area (data from SMF_ADM_AREA; see the section "Setting up the FT parameter library" on page 49)

Layout of the user information

Offsets (hex.)		**Length** (dec.)	**Format**	**Description**
abs	rel			
54	00	8	A	User ID from the TRANSFER-ADMISSION
5C	08	40	C	"accounting information" from the TRANSFER-ADMISSION
84	30	8	A	User ID of the user who submitted the request (only for transfer requests submitted in the local system)

Layout of the basic information

Offsets (hex.)		Length (dec.)	Format	Description
abs	rel			
8C	00	12	Z	Time when the file transfer request was stored, in the format YYMMDDhhmmss (applies only to requests issued in the local system)
98	0C	12	Z	Time when the transfer ended, format YYMMDDhhmmss
A4	18	1	C	Result of the transfer: + : successful transfer, - : unsuccessful transfer
A5	19	1	C	Follow-up processing in the local system + : was started - : was not started 0 : was not specified
A6	1A	8	A	Name of the remote system
AE	22	1	A	Transfer request was submitted L : in the local system, R : in the remote system
AF	23	11	Z	Transfer ID
BA	2E	2	-	Reserved
BC	30	4	B	Number of disk accesses (6)
C0	34	4	B	Number of bytes on disk (7)
C4	38	4	B	Number of bytes in network (8)

(6) Restrictions:
- For VSAM files, the number of control intervals is specified instead of the number of times the disk is accessed.
- When PO members are written with *EXTEND, a copy is first made of the old member. This counts as 1 disk access only.
- If restarts take place during file transfer, slight inaccuracies may occur in determining the number of times the disk is accessed.

(7) For files with record format V (or, in the case of VSAM files, where MAXLRECL is not equal to the AVGLRECL), the number of data bytes plus four times the number of records is specified. For VSAM files, the bytes reserved for the control interval definition fields and record definition fields are not taken into account.

(8) Number of bytes transferred, according to the table "Number of transferred bytes in the result list" in the section "Result lists produced by openFT for z/OS" in the User Guide "openFT for z/OS - Managed File Transfer in the Open World".

Layout of the file information

Offsets		**Length**	**Format**	**Description**
(hex.)		(dec.)		
abs	rel			
C8	00	2	B	Length of the file name
CA	02	2	-	Reserved
CC	04	See below	F	File name (9)

(9) The length of this data field is specified in the data field "length of the file name" (maximum length: 56).

9.4 The openFT job log

The openFT job log contains the following information:

- z/OS messages caused by openFT, e.g.:

  ```
  IEC130I OPFTPARM DD STATEMENT MISSING
  IEF212I ........ ...... OPFTPARM - DATA SET NOT FOUND
  IEF212I ........ ...... DDUADS - DATA SET NOT FOUND
  ```

  (see the description of the FJGEN command, page 182).

  ```
  CSV003I REQUESTED MODULE OPENFTCR NOT FOUND
  ```

  (see section “Installation of the openFT-CR delivery unit” on page 42)

  Which of these system messages are actually displayed also depends on your system environment.

- FT administration commands and the associated openFT synchronous messages:

  – FT administration commands which were entered at an operator console (possibly under NetView); these are indicated by an arrow "===>", e.g.:

    ```
    ===> FTSHWOPT
    ```

  – FT administration commands which were entered from TSO sessions; these are indicated by an arrow in the form "+++>", e.g.:

    ```
    +++>  FTSTART************
    FTR0500 OPENFT: openFT 11.0A00 starting. Protocols: openFT,FTP,ADM
    ```

  In the case of FT administrator commands whose names begin with FT, no parameters are output in the openFT job log, regardless of how many parameters were entered, e.g. the command FTSHWLOG is displayed as follows:

  ```
  +++> FTSHWLOG ************
  ```

  or

  ```
  ===> FTSHWLOG ************
  ```

  Synchronous messages issued by openFT in response to FT administration commands which were entered at an operator console (possibly under NetView) also are output by z/OS at the start of the job log, preceded by a plus sign "+".

- Asynchronous openFT messages:

  If asynchronous messages occur, they are always recorded in the openFT job log, regardless of whether

  - they are output at a TSO terminal at which FT administration mode is switched on (or they are collected for output at this type of terminal), or
  - they are output additionally or exclusively at one or more consoles; in this case, they also appear at the start of the job log, preceded by a plus sign "+".

- RACF messages concerning the rejection of checks in the context of transfer requests.
- Asynchronous messages output to one or more consoles on successful/unsuccessful file transfer (see keywords SUCC_MSG, FAIL_MSG and ENDMSG_ROUTCDE in the PARM member of the FT parameter library, starting on page 58), e.g.:

  ```
  12.01.56 JOB12345 FJM2100 FILE TRANSFERRED, TRANS_ID 1234567890
  ```

- Asynchronous messages output to a TSO terminal after a file transfer (see ENDMSG_TO_TSO in the PARM member of the FT parameter library, page 58ff), e.g.:

  ```
  12.01.56 JOB12345 SE 'FTR0005 OPENFT:Request 1234567890. File
  "DATASET.TEST" transferred, USER=...
  ```

- A list of all file-specific character sets used by openFT because of the specifications made in the member FNAMECTB of the FT parameter library (see page 82).

9.5 Reporting errors

The measures to be taken when an error message occurs are described under the message involved (see page 430).

The tips given in the previous section should help you rectify faults or bottlenecks which occur during FT operation.

If serious errors occur that lead to openFT terminating with a dump, the following information may be of help when trying to find the reason for the error: If a user abend code with a value below 4094 is reported, the code corresponds to a system abend code in decimal presentation (e.g. user abend code 1667 equals system abend code 683).

As an FT administrator, you must also advise FT users who are in doubt or who cannot rectify certain errors themselves. The section "Hints for the FT user" in the User Guide "openFT for z/OS - Managed File Transfer in the Open World" can help you in this case.

If, despite taking every precaution, an error occurs that neither you nor the system administrator can resolve, please contact your Service Center. In order to simplify diagnostics, you should supply the following documentation:

- a precise description of the circumstances in which the error occurred and whether it can be reproduced;
- the openFT trace files, see page 419'. If possible, the trace files should be formatted with the FTTRACE command, and, if applicable, the FT trace from the remote FT system. Run trace to cover a longer period (>= 2 h) in order to provide sufficient time stamps and possibly connection cleardown or, in the case of reproducible errors, activate trace **before** reproducing the same error);
- if available, the command call and result list of the request that produced the error;
- openFT job log;
- general documentation similar to that provided for system errors:
  - type of system (z/OS,...) and system version,
  - name and version of the job entry subsystem installed (JES2, JES3, ...),
  - information about the data protection support installed with name and version (SYS1.UADS, RACF, TOP-SECRET, ACF-2, MVS router exit, openFT-AC),
  - version of the Data Facility Product (DFP) installed, if applicable,
  - openFT version installed,
  - complete list of openFT corrections used;
- details of the FT partner and the transport system used (e.g. DCAM, CCP / CMX, VTAM, TRANSIT, etc.);

- if applicable, openFT dump:
  To obtain additional diagnostics information, you should populate the DIAGPAR member of the FT parameter library and attempt to create the openFT dump again. See also page 141.

The versions of the openFT modules installed can be determined using the FT administration command FJVERS (see page 422).

9.6 Diagnostic aids

Traces:

openFT	`FT-Trace`	supplies information about the NEABF protocol.
`GTF trace`	`including`	supplies information about the progress of SVC trace execution.
`VTAM`	`buffer trace`	supplies information about data traffic between VTAM and VTAM application.
`VTAM`	`line trace`	supplies information about data traffic via the line (SDLC protocol).

Obtaining information:

openFT	NSTATUS command indicates the status of requests.
`VTAM`	DISPLAY command indicates status of local LUs and PUs
`NETVIEW`	indicates the status of the network.
`NETSTAT`	provides information about the TCP/IP network.

9.6.1 FTTRACE - Convert trace data to readable form

Trace data which has been generated using the trace function (see FTMODOPT command, TRACE operand, page 254) can be converted into a readable form using the FTTRACE command. Before issuing this command, you must deactivate the trace function. This command can only be entered in TSO command mode.

9.6.1.1 Format of the trace files

openFT writes trace data to files with the following format:

- '<openft qualifier>.<inst>.Smddhhmm.Sssccc.I000.FTTF'
  (Control process)
- '<openft qualifier>.<inst>.Smddhhmm.Sssccc.Iiii..FTTF'
  (Server process for inbound and asynchronous outbound requests, i= 001,002, ...)
- '<openft qualifier>.<inst>.Ymddhhmm.Sssccc.Pnnnnnnn.FTTF'
  (Process for synchronous outbound requests)

Here, the first two name parts are replaced by OPENFT QUALIFIER and the name of the instance.

mddhhmm.Sssccc specifies the creation time of the trace file. Here, m indicates the month (1 = January, 2 = February, ... A= October, B=November, C = December), dd the day, hhmm the time in hours (hh) and minutes (mm), ssccc the time in seconds (ss) and milliseconds (ccc). nnnnnnn means the process ID of the process for synchronous outbound requests.

Please note that the trace file name may be shortened if the OPENFT QUALIFIER does not consist solely of a "first level qualifier", i.e. it contains a period. For example, ssccc may replaced by sss or may be omitted completely.

Trace files in the event of errors

- If a trace file cannot be written without errors due to a memory bottleneck, a DLOG record and a console message are output.
- If a record of the trace file cannot be written as a result of an infringement of the maximum record length, the trace file is closed and the subsequent records are written to a new continuation file with the additional suffix.Liii, e.g.:
  ’<openft qualifier>.<inst>.S8101010.S33222.I001.FTTF’ (first trace file)
  ’<openft qualifier>.<inst>.S8101010.S33222.I001.L001.FTTF’ (continuation file)

9.6.1.2 FTTRACE command

The FTTRACE command writes the converted trace to SYSPRINT, which is normally directed to the TSO console.
To write the converted trace to a file, the file must first be allocated, e.g.:

- FREE DDNAME(SYSPRINT)
- ALLOC DSNAME(TEST.TRACOUT) DDNAME(SYSPRINT) NEW CATALOG
- FTTRACE STD.S3141220.S44944.P3473434.FTTF
- FREE DDNAME(SYSPRINT)
- ALLOC DSNAME(*) DDNAME(SYSPRINT)

For large traces, sufficient storage space must be provided in the ALLOC command using the SPACE parameter.

Format

```
fttrace  -h |
        [ -d ]
        [ -sl=n | -sl=l | -sl=m | -sl=h ]
        [ -cxid=<context id> ]
        [ -f=hh:mm:ss ]
        [ -t=hh:mm:ss ]
          <trace files>
```

Description

-h Outputs the command syntax on screen. Any specifications after *-h* are ignored.

-d Specifies that the trace files are to be output in hexadecimal format (dump format).

If you do not specify *-d* then the files are output in printable form, default value.

-sl=n | **-sl=l** | **-sl=m** | **-sl=h**
Specifies the security level for the output.

n (no) No security requirements, i.e. all data is output including IDs, passwords, file names etc.

l (low) Passwords are overwritten with XXX.

m (medium)
Passwords, user IDs, account numbers, and follow-up processing commands are overwritten with XXX, default value.

h (high)
Passwords, user IDs, account numbers, follow-up processing commands and file names are overwritten with XXX, default value.

-cxid=context id

Selects the trace entries on the basis of the context ID. This is made up as follows: the first character is the slot pool ID and the second to fourth characters are the ID of the slot. If you omit *-cxid* or specify *-cxid=* without a context ID then trace entries are output for all context IDs.

-f=hh:mm:ss (from)

Specifies the time as of which trace entries in the trace file are to be evaluated.
You enter the time in the format hours:minutes:seconds (2 digits each).
If you do not specify a start time then trace entries are output from the start of the file.

-t=hh:mm:ss (to)

Specifies the time up to which trace entries in the trace file are to be evaluated.
You enter the time in the format hours:minutes:seconds (2 digits each).
If you do not specify an end time then trace entries are output up to the end of the file.

trace files

Name(s) of the trace file(s) that you want to evaluate. You can specify multiple trace files and wildcards can be used.

Example

As an FT administrator, you want to create a trace file and then convert the data contained in this file into a readable form. You must take the following steps:

- switch on the trace function (in administration mode),
- switch off the trace function (in administration mode),
- convert the trace data into a readable form (in TSO command mode).

The commands you must enter are shown below:

```
READY
ftmodopt trace=*on
READY
.
   .                           (period during which the trace data
   .                            is being logged)
   .
READY
ftmodopt trace=*off
READY
fttrace std.S4051730.S13145.P1234567.FTTF
   (Trace data is output to screen)
READY
```

By default, FTTRACE outputs the data to the TSO console. If the data is to be output to file, you must allocate SYSPRINT accordingly before FTTRACE is called.

9.6.2 FJVERS - Display openFT load module versions

The FJVERS command is used to display the versions of the installed openFT load modules. This command can only be entered in TSO command mode.

FJVERS

Example

If openFT-AC and openFT-FTP are installed, the output may look like this:

```
 READY
fjvers
 VERSION OF 'OPENFTAC' IN LIBRARY 'OPFTCHS.OPENFT.LOAD' IS '11.0A00 FTAC'
 VERSION OF 'OPENFTP' IN LIBRARY 'OPFTCHS.OPENFT.LOAD' IS '11.0A00'
 VERSION OF 'OPENFT' IN LIBRARY 'OPFTCHS.OPENFT.LOAD' IS '11.0A00'
 VERSION OF 'FTTRACE' IN LIBRARY 'OPFTCHS.OPENFT.NCLOAD' IS '11.0A00'
 VERSION OF 'NCOPY' IN LIBRARY 'OPFTCHS.OPENFT.NCLOAD' IS '11.0A00'
READY
```

9.6.3 FTSHWD - Display diagnostic information

The FTSHWD command outputs any diagnostic codes (together with date and time) that may have been written during the error event.

FTSHWD

Without operands

Example

FTSHWD supplies the following output:

```
FTSHWD
DATE           TIME      SSID  COMPONENT    LOCATION-ID         INFO
20090525       131251    FT    79/yfasdia   3/EuisyMsg          fd00000c
```

9.7 Internal openFT data sets

When certain FT administration commands are executed, openFT creates self-controlled internal data sets which are required for normal operation (logging file) or which contain diagnostic data (dump and trace files). These data sets must be deleted explicitly (dump files).

When using FTAC, openFT also automatically creates FTAC files in which the data generated and needed by FTAC are stored.

In total, the following internal openFT data sets exist for each openFT instance:

<openft qualifier>.<inst>.SYSRQF
Request queue (DA data set on disk)

<openft qualifier>.<inst>.SYSPTF
Partner list, corresponds to what used to be the network description file (DA data set on disk)

<openft qualifier>.<inst>.SYSOPF
Operational parameters file (DA data set on disk)

<openft qualifier>.<inst>.SYSLOG etc.
Components of the logging file:

<openft qualifier>.<inst>.SYSLOG: PS data set

<openft qualifier>.<inst>.SYSLOG.P00,
<openft qualifier>.<inst>.SYSLOG.P00.DATA,
<openft qualifier>.<inst>.SYSLOG.P00.INDEX: Components of a VSAM cluster

Instead of the second level qualifier <inst>.SYSLOG used as a standard, a value specified by the administrator can be used (keyword LOGFILE_2ND_Q in the member PARM of the FT parameter library, see page 52).

All FT and FTAC logging records are stored in that file. If the file is deleted or corrupted by individual components being deleted, the logging records are all lost.

If problems occur when the logging file is created or when it is being accessed, openFT cannot be started. (Such problems might arise from there being insufficient storage space or due to access protection for the file; refer to section “Protecting openFT administrative files” on page 33 for further information on admission protection.) The openFT job log file will contain the following message for example:

```
FTR0855 OPENFT: No space left on device for internal files
```

The FT system creates the logging file with the primary allocation, which you specified with the option LOGFILE_SIZE_RC (member PARM in the FT parameter library, see page 52); the value is halved for the size of the secondary allocation. The total size of the logging file depends on the number of logging records is contains. In your role as

FT administrator, depending on the volume of requests, you should save the existing logging records from time to time and then delete them from the system using FTDELLOG (see description of the command FTDELLOG on page 234). This preserves contiguous documentation of the logging record over a longer period of time, while at the same time freeing storage space. Note that the allocated file size does not change. The space no longer occupied within the file is simply released again.

We recommend you use the following command from time to time to check to what extent the file contents has been split up:

```
LISTCAT ENT('<openft qualifier>.<inst>.SYSLOG.P00') ALL
```

If the file contents is split up to much, performance may deterioration. In that case the file should be reorganized, i.e. a new VSAM cluster with the same characteristics as the existing one should be created and the file contents should be copied using REPRO.

If no further logging records can be written into the logging file because it is full, the openFT instance system automatically deactivates itself (with the internal execution of the command FTSTOP, see page 362). The openFT job log contains the system message IEC070I (meaning "An error occurred during EOV (end_of_volume) processing for a VSAM data set"). The FT administrator must then first make space available in the logging file by deleting logging records (command FTDELLOG, see page 234), then the FT system can be reactivated (FJSTART) and FT operation can continue.

<openft qualifier>.<inst>.SYSFSA etc.
Components of the FTAC file (only when FTAC is used):

<openft qualifier>.<inst>.SYSFSA: PS data set

<openft qualifier>.<inst>.SYSFSA.P00,
<openft qualifier>.<inst>.SYSFSA.P00.DATA,
<openft qualifier>.<inst>.SYSFSA.P00.INDEX,
<openft qualifier>.<inst>.SYSFSA.P01,
<openft qualifier>.<inst>.SYSFSA.X01,
<openft qualifier>.<inst>.SYSFSA.X01.DATA,
<openft qualifier>.<inst>.SYSFSA.X01.INDEX: Components of a VSAM cluster

Instead of the second level qualifier <inst>.SYSFSA used as a standard, a value specified by the administrator can be used (keyword FILE_2ND_Q in the member FTACPAR of the FT parameter library, see page 84).

The file contains the FTAC environment, i.e. the admission sets, admission profiles, etc. If the file is deleted or corrupted by individual components being deleted, all admission profiles and admission sets are lost.

If problems occur when the FTAC file is created or when it is being accessed, openFT cannot be started. (Such problems might arise from there being insufficient storage space or due to access protection for the file; refer to section “Protecting openFT administrative files” on page 33 for further information on admission protection.)
The openFT job log file will contain the following message for example:

```
FTR0855 OPENFT: No space left on device for internal files
```

The FT system creates the FTAC file with the primary allocation which you specified in the parameter FILE_SIZE_KB (member FTACPAR of the FT parameter library, see page 84); the value is halved for the size of the secondary allocation.

We recommend you use the following command from time to time to check to what extent the file contents has been split up:

```
LISTCAT ENT('<openft qualifier>.<inst>.SYSFSA.P00') ALL
```

If the file contents is split up to much, performance may deterioration. In that case the file should be reorganized, i.e. a new VSAM cluster with the same characteristics as the existing one should be created and the file contents should be copied using REPRO.

If an FTAC command with which data are to be stored in the FTAC file fails (e.g. the command FTCREPRF, see page 212) because the file is too full, the command is rejected and the following message is issued:

```
FTC0255 CMD TERMINATED. SYSTEM ERROR
```

The FT system automatically deactivates itself (with the internal execution of the command FTSTOP, see page 362). The openFT job log contains the system message IEC070I (meaning "An error occurred during EOV (end_of_volume) processing for a VSAM data set").The FT system only needs to be restarted (FTSTART) in order to continue FT operation. However, before any further information can be stored in the FTAC file, the FTAC administrator must make space available in the FTAC file by deleting admission sets and admission profiles that are no longer required.
The FTAC administrator can find out which admission profiles exist by having them displayed with the command FTSHWPRF (see page 346). Admission profiles are deleted with the command FTDELPRF (see page 238).
There is no special command for deleting admission sets. The FTAC administrator deletes an admission set by setting the admission set for the relevant user ID back to the standard admission set (command FTMODADS, see page 248, with MAX-LEVELS=*STD). This can also be done for user IDs that have already been deleted. The FTAC administrator can find out which user ID have an admission set that differs from the standard admission set with the command FTSHWADS (see page 304).

<openft qualifier>.<inst>.SYSFDF.Ddddmmmm
openFT dump file (PS data set on disk)
Dump information in this file is written automatically when a serious openFT error is encountered (e.g. protocol infringement, error situations where the messages FTR4024ff are issued, incorrect behavior of an openFT-specific exit routine).

'<openft qualifier>.<inst>.Smddhhmm.Sssccc.Iiii..FTTF" or
'<openft qualifier>.<inst>.Ymddhhmm.Sssccc.Pnnnnnnn.FTTF'
openFT trace file (FT trace file, PS dataset with 'Undefined' record format on disk).
(see the FTMODOPT command, page 254.)

Explanations

<openft qualifier>
OPENFT QUALIFIER that was defined using the FJGEN command (see page 191)

<inst>	Name of the openFT- instance
ddd	day of the year
mmmm	minute of the day
yymmdd	Date (year-month-day)
hhmmss	Time (hour-minute-second)

The request file, the partner list, the log file and the FTAC file are set up on the volume specified for this purpose in the PARM member of the FT parameter library (keywords NABVOLUME/NABUNIT). If no specification is made here, these files are set up on the same volume as the trace and dump files. You can define this volume, too, in the PARM member of the FT parameter library (keywords DMP_VOLUME/DMP_UNIT). If no specification is made here either, these files are set up on the volume specified for VOLUME/UNIT in the FJGEN command.

The dump and trace files are set up on the volume specified for this purpose in the PARM member of the FT parameter library (keywords DMP_VOLUME/DMP_UNIT). If no specification is made here, the dump and trace files are set up on the volume specified for VOLUME/UNIT in the FJGEN command.

9.8 Temporary openFT data sets

In order to execute certain functions, openFT creates temporary data sets. These are normally deleted automatically after the function has been executed. If, due to an error, they are retained, they must be deleted explicitly. The temporary openFT data sets are as follows:

transuid.podsname.U

Each time an entire PO or PDSE data set is transferred, a temporary PS data set is created in the send and receive system as a buffer for the file in "unloaded" format (IEBCOPY). These data sets are normally deleted after transfer.

transuid:	User ID specified in the TRANSFER-ADMISSION for the system involved.
podsname:	Partially qualified name of the PO or PDSE data set.
.U:	This suffix identifies the temporary PS data set.

These temporary PS data sets are set up on the volume specified for this purpose in the PARM member of the FT parameter library (keywords UNLOADVOL/UNLOADUNIT). If no specification is made here, the system defaults for newly created files apply.

Each of these temporary PS data sets has approximately the same storage requirements as the corresponding PO data set.

<openft qualifier>.IEBSPILL.ddn1.ddn2

Each time an entire PO or PDSE data set is transferred, small temporary PS data set is created in the send and receive system as a buffer for creating the directory in the event of a main memory bottleneck. These data sets are normally deleted after transfer.

<openft qualifier>:	OPENFT QUALIFIER that was defined using the FJGEN command (see page 191)
ddn1, ddn2:	DD name supplied by the z/OS system.

These scratch files are set up on the default volume (system-specific).

FJCMD.TMP.OUT

When some menu interface functions are executed, a temporary PS data set is created as a buffer for the command. This data set is normally deleted after execution of the function.

These scratch files are set up on the default volume (system-specific).

<openft qualifier>.<inst>.S.PP.@num.id.ERR
<openft qualifier>.<inst>.S.PP.@num.id.OUT

Scratch files that are created during preprocessing and postprocessing are usually then deleted again.

<openft qualifier>:
OPENFT QUALIFIER that was defined using the FJGEN command (see page 191)

<inst>: Instance name of the currently set openFT instance

9.9 FT system messages

The structure of the FT system messages is as follows:

```
FJMnnnn OPENFT: message text
```

or

```
FTCnnnn message text
```

FTRnnnn

is the message code. The message code is 7 characters long.

message text

is the message text. The text appears in uppercase letters. The message text can contain what are known as inserts, e.g. (&00). These parts of the messages are supplied with the current value (e.g. transfer ID) when the message is output.

Additional explanatory information for the message is given under "Meaning"; "Response" tells you what action you should take. The texts are not displayed with the message.

Messages with the message code **FTRnnnn** (nnnn < 4000) and **FTCnnnn** are displayed both for the FT user and for the FT administrator.

Messages with the message code **FTR4nnn** are only displayed for the FT administrator.

All message lists were generated with a view to your finding in them any error message that you might encounter. Consequently, the lists also contain a series of messages, that are only output under very specific circumstances (i.e. very rarely).

This also means that you cannot always expect the message from among those contained in the lists that would seem most appropriate to you.

Asynchronous messages for the FT administrator

Messages are normally a response to administration commands. There are, however, messages that are not generated by, or not only by, administration commands. These messages are output when FT administration mode is activated and/or to a console/a number of consoles and written to the openFT job log.

The meaning of these messages and the responses are explained on page 432.

Error codes and additional information

Additional error codes and supplementary partner-specific information can be output for some user messages. These provide additional information for troubleshooting.

As a rule, this supplementary information is made up of a return code from the operating system together with a text supplied by the operating system issued in the language set in the operating system. The supplementary information can also comprise an English language text generated by openFT itself.

It is appended to the end of the message and is restricted to a length of 64 characters. Longer texts are truncated.

9.9.1 FTR4nnn messages

FTR4004 `OPENFT: SMF NOT ACTIVE.`

Meaning
When an attempt was made to write an accounting record to the SMF file, SMF was found to be inactive. No more accounting records are written for any subsequent transfer requests.

FTR4005 `OPENFT: SMF ERROR.`

Meaning
An error (possibly a temporary one) occurred when writing an accounting record.

FTR4006 `OPENFT: SMF RECORDING STOPPED.`

Meaning
No more accounting records are written for any subsequent transfer requests because errors occurred during 10 consecutive attempts to write a record to the SMF file or SMF was found to be inactive or no longer active.

FTR4010 `OPENFT: INQUIRE FOR COMMON BUFFER SPACE FAULTY`

Meaning
Not enough memory could be reserved for buffering a command entered at an operator console (asynchronous message issued to the FT administrator).

Response
Ask the system administrator.

FTR4026 `OPENFT: CONSOLE–TASK EVENTING ERROR`

Meaning
The console task has received an unexpected event. (This error message only appears in the openFT job log. When this error occurs, a dump is generated and written to the file SYSFDF.Ddddmmmm. openFT continues to execute, however.)

Response
Ask the system administrator.

FTR4040 `OPENFT: UNABLE TO OPEN TNSTCPIP.`

Meaning
The TNSTCPIP member of the FT parameter library could not be opened.

Response
If openFT is to be connected to remote systems via TCP/IP: Make sure that the FT parameter library and TNSTCPIP member both exist and that openFT can access them.

FTR4041 `OPENFT: ERROR OCCURRED WHILE READING TNSTCPIP.`

Meaning
An error occurred while the TNSTCPIP member was being read in from the FT parameter library.

Response
If openFT is to be connected to remote systems via TCP/IP: Make sure that the FT parameter library and TNSTCPIP member both exist and that openFT can access them.

FTR4042 `OPENFT: SYNTAX ERROR IN TNSTCPIP ENTRY.`

Meaning
A syntax error was discovered in one of the data records in the TNSTCPIP member of the FT parameter library. The defective data record is ignored and reading of the TNSTCPIP member is continued. The field in which the syntax error was first identified is generally indicated by means of one of the messages below (FTR4043 to FTR4046).

Response
Correct the syntax error (see additional message) and enter the FTUPDPAR command.

FTR4043 `OPENFT: ILLEGAL TNS-NAME (&00).`

Meaning
Illegal syntax has been used for the TNS name (&00) (additional message to message FTR4042).

Response
Correct the TNS name (name of address entry). A valid TNS name (name of the address entry) must be unique within the local system and consist of a maximum of 8 alphanumeric characters, the first of which must be a letter or one of the special characters $, @ or #.

FTR4044 `OPENFT: ILLEGAL INTERNET ADDRESS (&00).`

Meaning
Illegal syntax has been used for the Internet address (&00) (additional message to message FTR4042).

Response
Correct the Internet address. A valid Internet address has the form xxx.xxx.xxx.xxx, where xxx is an integer (in decimal representation) in the range 0 to 255.

FTR4045 `OPENFT: ILLEGAL PORT NUMBER (&00).`

Meaning
Illegal syntax has been used for the port number (&00) (additional message to message FTR4042).

Response
Correct the port number. A valid port number consists of an integer in the range 1 to 32767.

FTR4046 `OPENFT: ILLEGAL TSEL (&00).`

Meaning
Illegal syntax has been used for the T-selector (&00) (additional message to message FTR4042).

Response
Correct the T-selector. A valid T-selector can consist of up to 32 characters.

FTR4048 `OPENFT: TNSTCPIP RECORD LIMIT EXCEEDED.`

Meaning
The TNSTCPIP member contains more than 10000 records. All records after record number 10000 are ignored.

Response
This message can be ignored if you are sure that the specifications for all partner systems that are to be accessed via TCP/IP are contained in the first 10000 records of the member.

FTR4053 `OPENFT: CONNECTION TO TCP/IP SEVERED. REASON = (&00)`

Meaning
The openFT connection to TCP/IP activated previously (see message FTR4051) has been aborted. File transfers via TCP/IP are no longer possible.
Either a detailed reason is given as reason code in this message or the original reason code of the software product used for the link to TCP/IP (TCP/IP (for MVS) from IBM or compatible product).

Response
First, the cause of the unwanted termination of the software product used for the TCP/IP link must be removed.
Since openFT itself does not try to restore the connection to TCP/IP, the FT system must then be deactivated (FTSTOP command) and reactivated again (FTSTART command).

FTR4054 `OPENFT: MYPORT-NUMBER (&00) NOT AVAILABLE`

Meaning
A value for the openFT passive port that is already used by another TCP application was specified in the PORT parameter of the FJGEN command.

Response
Either terminate the TCP application that is occupying this port number or deactivate openFT (FTSTOP), correct the PORT specification in the FJGEN command and reactivate openFT (FTSTART).

FTR4055 `OPENFT: TCP/IP-TRANSPORT NOT ENABLED. REASON=(&00)`

Meaning
openFT was unable to log on to the TCP/IP transport system. The TCP/IP address space cannot be accessed.

Response
Check the generation for connecting openFT to TCP/IP. You may have to specify or check the name of the TCP/IP address space (TCP_USERID in the PARM member of the parameter library). If you are unable to find the reason for the error, report the error and the reason (&00) contained in the message to your Service Center.

FTR4056 `OPENFT: HOST NAME (&00) UNKNOWN`

FTR4057 `OPENFT: LOCAL IP-ADDRESS (&00) NOT SUPPORTED`

FTR4120 `OPENFT: INITIATED`

Meaning
The openFT instance has been loaded in response to the FJINIT command.

FTR4121 `OPENFT: TERMINATED`

Meaning
The openFT instance has been unloaded in response to the FTTERM command, or abnormally terminated due to a serious error.

FTR4125 `OPENFT: PARAMETERS TNSTCPIP, FTADM and FTACADM UPDATED`

Meaning
The parameters have been successfully updated using the FTUPDPAR command.

FTR4131 `OPENFT: TERMINATION INITIATED BY USER`

Meaning
A user has entered the FTTERM command under TSO.

FTR4140 `OPENFT: TERMINATED. MAX NO. OF INSTANCES EXCEEDED.`

Meaning
An attempt was made to load a 17th openFT instance.

Response
Terminate another instance with FTTERM or Cancel.

FTR4141 `OPENFT: TERMINATED. INSTANCE NAME IN USE.`

Meaning
An attempt was made to load openFT with an instance name that is already in use.

Response
Use a different instance name.

FTR4144 `OPENFT: CMD REJECTED. USER NOT AUTHORIZED`

Response
An NCOPY command was entered by a job for which there is neither a user ID ("user-id.") or a "dsname prefix" or a command was entered that the caller does not have permission to issue.

FTR4145 `OPENFT: CMD REJECTED. SESSION IDENTIFICATION FAILED`

FTR4150 `OPENFT: CMD REJECTED. SYNTAX ERROR`

Meaning
This message is output if the command is entered with incorrect syntax (e.g. command name written wrongly).

FTR4180 `OPENFT: CMD REJECTED. INTERNAL SYSTEM CALL FAILED`

FTR4192 `OPENFT: (&00) NOT KNOWN TO TRANSPORT SYSTEM`

Meaning

a) An application, LU or LOGMODE name (&00) was not found in the generation of the transport system (VTAM). If this message is issued for an NCOPY/NCANCEL/NSTATUS command (&00 = FJNNDMS0, FJNNDMS1,...), the specified LU has not been generated or all generated LUs of this type are currently reserved.

b) The attempt to determine the Internet address of a remote computer from its host name (&00) via the z/OS Name Services, i.e. either via the "Domain Name System" (DNS) or the file TCPIP.HOSTS.LOCAL ("Flat Name Space"), has failed.

FTR4193 `OPENFT: (&00) NOT AVAILABLE`

Meaning
An application or openFT is currently not available. If this message is issued during processing of an NCOPY/NCANCEL/NSTATUS command and (&00)=LU, all LUs of the type FJNNDMSx are generated and currently reserved. If this message is issued during processing of an NCOPY/NCANCEL/NSTATUS command and (&00)=FJNNDMSx, a VTAM open error has occurred for this LU.

FTR4196 `OPENFT: DIALOG HANDLER VERSION INCOMPATIBLE`

FTR4197 `OPENFT: (&00) TERMINATED BY TIMEOUT`

FTR4199 `OPENFT: SYSTEM ERROR. ERROR CODE (&00)`

Meaning
An operating system function called by openFT has reported an error. The error code (&00) has two halves:

left half	operating system function used.
right half	return code of the operating system function used.

FTR4200 `OPENFT: FTADM VERSION (&00) INITIATED.`

9.9.2 FTR messages

FTR0000 `OPENFT: Request (&00) accepted.`

Meaning
The command has been stored in the local system's request queue. File transfer will begin once all the resources have been assigned in both the local and remote system.

(&00): transfer ID assigned by the local FT system. You need the transfer ID in case you wish to cancel (NCANCEL) the FT request later.

FTR0005 `OPENFT: Request (&00). File '(&01)' transferred.`

Meaning
The file transfer request (&00) has been completed successfully. Follow-up processing for both the local and remote system, if requested, has been initiated (provided no error occurred). Local Errors are indicated by a message.

FTR0020 `OPENFT:'(&00)' not found.`

Meaning
The command has not been executed because the send file is not cataloged or not on a volume of the local system. The command has not been executed because either the send file is not/is no longer, or the receive file is no longer in the catalog or on a volume of the relevant system.

Response
Correct the file name, read in file from tape or restore send file. Repeat the command.

FTR0035 `OPENFT: File locked to prevent multiple access.`

Meaning
The command has not been executed because either the send file or the receive file is already locked by another process against simultaneous updating.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files that are not closed correctly.

FTR0041 `OPENFT: Request queue full.`

Meaning
The command has not been executed because the maximum number of permissible transfer requests has been reached.

Response
Notify the FT administrator. Repeat the command later.

FTR0108 `OPENFT: Request (&00). Remote system not accessible.`

Meaning
The command could not be accepted because the partner system is currently not available.

Response
Repeat the command later. If the error persists, contact the system or network administrator.

FTR0236 `OPENFT: Current instance (&00) no longer found`

Meaning
The command was rejected. The instance (&00) could not be found.

FTR0301 `OPENFT: Partner '(&00)' entered state NOCON.`

Meaning
The partner system (&00) has switched to the state NOCON. This state means that the partner is no longer accessible.

Response
If necessary, check whether the connection to the partner system has been interrupted.

FTR0302 `OPENFT: Partner '(&00)' entered state ACTIVE.`

Meaning
The partner system (&00) has switched to the state ACTIVE.

Response
For information only.

FTR0303 `OPENFT: Partner '(&00)' entered state LUNK.`

Meaning
The partner system (&00) has switched to the state LUNK. This state means that the local FT system is not known in the remote FT system.

Response
Ask the remote system’s FT administrator to enter the local system in the remote system’s network description file/partner list.

FTR0304 `OPENFT: Partner '(&00)' entered state RUNK.`

Meaning
The partner system (&00) has switched to the state RUNK. The state RUNK means that the remote system is not known in the local transport system.

Response
Make the remote system known on the local system.

FTR0305 `OPENFT: Partner '(&00)' entered state INACT.`

Meaning
The partner system (&00) has switched to the state INACT. The state INACT means that the FT administrator has locked outbound requests for this partner system.

Response
Remove the lock if necessary.

FTR0306 `OPENFT: Partner '(&00)' entered state AINACT.`

Meaning
The partner system has switched to the state AINACT. The state AINACT means that the partner system has been automatically deactivated because a certain number of consecutive connection attempts have failed.

Response
Check whether partner system should be accessible and reactivate the partner system.

FTR0307 `OPENFT: Partner '(&00)' may be unreachable.`

Meaning
A number of consecutive attempts to connect to the partner system (&00) have failed. Further attempts will be made.

Response
For information only.

FTR0308 `OPENFT: Partner '(&00)' does not allow more inbound requests.`

FTR0309 `OPENFT: Partner '(&00)' added.`

Meaning
The specified remote system has been entered in the partner list.

FTR0310 `OPENFT: Partner '(&00)' removed.`

Meaning
The specified remote system has been removed from the partner list.

FTR0311 `OPENFT: Partner '(&00)' entered state LAUTH.`

Meaning
The partner system (&00) has switched to the state LAUTH. The state LAUTH means that the local system could not authenticate itself at the remote system.

Response
Send the current key file to the administrator of the remote system.

FTR0312 `OPENFT: Partner '(&00)' entered state RAUTH.`

Meaning
The partner system (&00) has switched to the state RAUTH. The state RAUTH means that the remote system could not authenticate itself at the local system. This may either be due to an out-of-date key in the key file or to may indicate an access attempt by an unauthorized system.

Response
Contact the administrator of the remote system.

FTR0313 `OPENFT: Partner '(&00)' entered state DIERR.`

Meaning
The partner system (&00) has switched to the state DIERR. File integrity errors have been detected on the transmission path. This may also indicated deliberate manipulation of the transmission data.

FTR0314 `OPENFT: Partner '(&00)' entered state NOKEY.`

Meaning
The partner system (&00) has switched to the state NOKEY. The state NOKEY means that the partner will not accept a connection without encryption or that no key is present.

Response
Generate a new key pair.

FTR0315 `OPENFT: Partner '(&00)' entered state IDREJ.`

Meaning
The partner system (&00) has switched to the state IDREJ. The local identification was not accepted by the local identification or by an intermediate entity.
Possible causes:

- both the local identification and the migrated ID %.<processor>.<entity> are entered in the remote system’s request file.
- the identification has been rejected by an intermediate entity for security reasons

Response
Ask for your entity’s partner entry to be checked.

FTR0320 `OPENFT: Abnormal termination initiated.`

Meaning
Abnormal termination of FT has been initiated due to an internal error.

Response
Check the cause of the abnormal termination and restart FT.

FTR0330 `OPENFT: Request queue 85 percent full.`

Meaning
Approximately 85% of the spaces for request storage in the request file are occupied. Issuing a number of additional requests could completely fill the request queue with the result that FT will reject new requests.

Response
If necessary, increase the size of the request queue.

FTR0331 `OPENFT: At least 20 percent of request queue unoccupied.`

Meaning
At least 20% of the FT request queue is available. This message is only output if a previous FTR0330 message has warned of a possible queue overflow. The threat of a bottleneck has receded.

FTR0340 `OPENFT: Transfer '(&00)' successfully completed.`

Meaning
The request designated in greater detail by the insert (&00) has been terminated successfully.
(&00): *LOC/*REM;SID;PARTNER;USERID;FILE
Since the length of the insert is limited to a maximum of 180 characters, the file name may be truncated if necessary. This is indicated by the character ’*’ at the end of the file name.

Response
For information only.

FTR0341 `OPENFT: Transfer '(&00)' terminated with error.`

Meaning
The request designated in greater detail by the insert (&00) terminated with an error
(&00): MSGNR;*LOC/*REM;SID;PARTNER;USERID;FILE
Since the length of the insert is limited to a maximum of 180 characters, the file name may be truncated if necessary. This is indicated by the character ’*’ at the end of the file name.

Response
For information only.

FTR0360 `OPENFT: openFT control process started`

Response
For information only.

FTR0361 `OPENFT: openFT control process terminated`

Response
For information only.

FTR0500 `OPENFT: openFT (&00) started. Protocols: (&01).`

Meaning
The openFT file transfer system openFT has been activated for the protocols (&01).

FTR0501 `OPENFT: openFT terminated.`

Meaning
The file transfer system openFT has been terminated by means of an administration command.

FTR0502 `OPENFT: No log records available for the selection criteria.`

Meaning
No logging records meet the selected criteria.

Response
Change the selection criteria.

FTR0503 `OPENFT: No partner available for the selection criteria.`

Meaning
There are no partners that meet the specified selection criteria.

Response
Change the selection criteria.

FTR0504 `OPENFT: No requests available for the selection criteria.`

Meaning
There are no requests that meet the specified selection criteria.

Response
Change the selection criteria.

FTR0505 `OPENFT: Requests carried out; (&00) files were transferred`

Meaning
The file transfer requests have been successfully completed. A total of (&00) files have been transferred. If you have specified commands for follow-up processing, follow-up processing is carried out for every file.

FTR0560 `OPENFT: Cancel all specified requests? Reply (y=yes; n=no)`

Meaning
A CANCEL-TRANSFER command applies to more than one file transfer.
Y: All the transfer requests affected are deleted.
N: The entire deletion request is withdrawn.

FTR0562 `OPENFT: (&00):`

FTR0600 `OPENFT: Shutdown processing delayed. FT tasks pending.`

Meaning
openFT could not be terminated.

Response
Check if there are console messages that need to be answered for FT tasks connected to the FT subsystem.

FTR0604 `OPENFT: Request (&00). Follow-up processing not started.`

Meaning
The follow-up processing of a transfer request was not started because the local processing admission may be incorrect.

Response
Correct the local processing admission and repeat the command.

FTR0605 `OPENFT: Tracefile changed`

Meaning
There has been a switch to a new trace file.

FTR0606 `OPENFT: Trace terminated.`

Meaning
The trace status has been switched off.

FTR0607 `OPENFT: Trace started: (&00).`

Meaning
The trace status for the protocols specified in (&00) has been switched on.

FTR0700 `Parameter '(&00)' and '(&01)' must not be specified at the same time`

Meaning
The selected parameters could not be specified simultaneously.

Response
Omit one of the two parameters and repeat the command.

FTR0701 `OPENFT: Input error`

FTR0702 `OPENFT: Parameter value '(&00)' too long`

Meaning
The specified parameter value (&00) is too long; see the command syntax.

Response
Reduce the length of the parameter value (&00) and repeat the command.

FTR0703 `OPENFT: Mandatory parameter missing`

Meaning
A mandatory parameter is missing; see the command syntax.

Response
Correct the command and try again.

FTR0704 `OPENFT: Mandatory parameter '(&00)' missing`

Meaning
The mandatory parameter (&00) was not specified.

Response
Correct the command and try again.

FTR0705 `OPENFT: Parameter '(&00)' specified more than once`

Meaning
The parameter (&00) was specified more than once.

Response
Correct the command and try again.

FTR0706 `OPENFT: Parameter '(&00)' can only be specified together with '(&01)'`

Meaning
The parameter (&00) can only be specified together with (&01).

Response
Add the parameter (&01) to the command and repeat the command.

FTR0707 `OPENFT: Invalid parameter '(&00)'`

Meaning
An invalid parameter (&00) was specified; see the command syntax.

Response
Correct the command and try again.

FTR0708 `OPENFT: Value of parameter '(&00)' not within valid range`

Meaning
The parameter value (&00) is not within the specified value range; see the command syntax.

Response
Correct the parameter value (&00) and repeat the command.

FTR0709 `OPENFT: Too many positional parameters`

Meaning
The maximum number of positional parameters was exceeded.

Response
Correct the command and try again.

FTR0710 `OPENFT: Invalid parameter value '(&00)'`

Meaning
The assigned parameter value (&00) is incorrect; see the command syntax.

Response
Correct the parameter value (&00) and repeat the command.

FTR0750 `OPENFT: Command not found`

FTR0751 `OPENFT: Command name ambiguous with regard to '(&00)'`

FTR0752 `OPENFT: Closing parenthesis missing for operand '(&00)'`

FTR0753 `OPENFT: Invalid delimiter '(&00)' after operand '(&00)'`

FTR0755 `OPENFT: List value of operand '(&00)' is not consistent with data type '(&00)'`

FTR0756 `OPENFT: Operand value introducing the structure is mandatory for '(&00)'`

FTR0757 `OPENFT: Value of operand '(&00)' is not consistent with data type '(&00)'`

FTR0758 `OPENFT: Keyword value of operand '(&00)' is ambiguous with regard to '(&00)'`

FTR0759 `OPENFT: Too many closing parentheses`

FTR0760 `OPENFT: The mandatory operand '(&00)' is missing`

FTR0762 `OPENFT: Operand name '(&00)' ambiguous with regard to '(&00)'`

FTR0763 `OPENFT: Operand '(&00)' is not known`

FTR0764 `OPENFT: Operand '(&00)' specified more than once`

FTR0765 `OPENFT: Too many list elements for operand '(&00)'`

FTR0766 `OPENFT: Too many positional operands`

FTR0767 `OPENFT: Too many positional operands for '(&00)'`

Meaning
(applies to FTR0750 through FTR0767)
An operand value that introduces a structure can only be omitted if there is only one possible structure specification for the corresponding operand or if this structure specification is the default value for the operand.
The following command, for example, will be rejected with this message:

```
   FTMODPRF MYPROF01,PARTNER=((REMSYS1,REMSYS2))
```

Reason: It is not clear which of the following specifications is meant:

```
   FTMODPRF MYPROF01,PARTNER=*ADD((REMSYS1,REMSYS2))
```

or

```
   FTMODPRF MYPROF01,PARTNER=*REM((REMSYS1,REMSYS2))
```

Response
Repeat the command using the correct syntax.

FTR0780 `OPENFT: Internal error: operand buffer overflow`

FTR0781 `OPENFT: Internal error: structure nesting too deep`

FTR0790 `OPENFT: Available commands: '(&00)'`

FTR0791 `OPENFT: Available list-values: '(&00)'`

FTR0792 `OPENFT: Available operands: '(&00)'`

FTR0793 `OPENFT: Available values: '(&00)'`

FTR0801 `OPENFT: Request (&00). Internal error`

Meaning
NDMS, FJAM or operating system error that is neither a DMS error nor a transport system error, possibly the transfer ID.
The FT system continues to run after the message has been issued.

FTR0804 `OPENFT: Request (&00). Request data inconsistent.`

FTR0851 `OPENFT: Internal error.`

FTR0852 `OPENFT: Internal error. Current instance '(&00)' incompatible.`

Meaning
The system data was not created with the version of the openFT file transfer system currently in use.

Response
Update the instance to the current openFT version using the appropriate command (FJGEN).

FTR0854 `OPENFT: Writing of log records no more possible. Process terminated.`

Meaning
There is not enough space on the disk/partition on which the logging files are stored.

Response
Increase the disk space (or have it increased).

FTR0855 `OPENFT: No space left on device for internal files.`

Meaning
There is not enough space on the disk/partition on which the internal files are stored.

Response
Increase the disk space (or have it increased).

FTR0856 `OPENFT: Error during ops generation.`

FTR0857 `OPENFT: Error in key file (&00)`

FTR0858 `OPENFT: Internal error. Set / release file-locks not possible`

Meaning
A problem occurred when setting/resetting the file locks for all open requests in FT-REQUEST-FILE.

Response
Check whether the request file SYSRQF is accessible on the config user ID of the current instance.

FTR0862 `OPENFT: Protocol stack (&00) not installed`

Meaning
The required transfer protocol is not installed.

Response
Install the transfer protocol.

FTR0863 `OPENFT: FTAC subsystem not available`

Meaning
Install openFT-AC.

FTR0999 `OPENFT: openFT panic (&00). Abnormal termination`

FTR1020 `OPENFT: openFT already started.`

Meaning
openFT can only be started once in each instance.

Response
Terminate openFT if necessary.

```
FTR1021     OPENFT: Request must be canceled without FORCE option first
```

Meaning
Before the FORCE option is used, the command must be called without the FORCE option.

Response
Issue the command without the FORCE option first.

```
FTR1029     OPENFT: Maximum number of key pairs exceeded.
```

Meaning
The maximum number of key pair sets has been reached.

Response
Before new key pair set can be created, an older key pair set must be deleted.

```
FTR1030     OPENFT: Warning: last key pair deleted.
```

Meaning
The last key pair set has been deleted. Without a key pair set, encrypted transfer, authentication and data integrity checking are not possible.

Response
Create a new key pair set.

```
FTR1031     OPENFT: No key pair available.
```

Meaning
All transfers are carried out without encryption.

Response
Create a new key pair set, if necessary.

```
FTR1032     OPENFT: Last key pair must not be deleted
```

```
FTR1033     OPENFT: The public key files could not be updated.
```

Meaning
The contents of the SYSPKF file could not be fully updated.
Possible reasons:

- The SYSPKF file is locked.
- There is not enough disk space to allow the file to be created.

Response
Take the appropriate action depending on the cause of the error:

- Unlock the file.
- Allocate disk space or have your system administrator do it.

Update the key with UPDATE-FT-PUBLIC-KEY.

FTR1034 `OPENFT: Command only permissible for FT or FTAC administrator`

Meaning
Only the FT or FTAC administrator is permitted to use the command.

Response
Have the command executed by the FT or FTAC administrator.

FTR1035 `OPENFT: Command only permissible for FT administrator.`

Meaning
Only the FT administrator is permitted to use the command.

Response
Have the command executed by the FT administrator.

FTR1036 `OPENFT: User not authorized for other user Ids.`

Meaning
The user is not authorized to use a different user ID in the command.

Response
Specify your own ID, or have the command executed by the FT or FTAC administrator.

FTR1037 `OPENFT: Key reference unknown.`

Meaning
The specified key reference is unknown.

Response
Repeat the command with an existing key reference.

FTR1038 `OPENFT: Request '(&00)' is in the termination phase and can no longer be canceled`

FTR1039 `OPENFT: openFT not active.`

Meaning
openFT is not started.

Response
Start openFT, if necessary.

FTR1040 `OPENFT: Config user ID unknown or not enough space`

Meaning
The CONFIG USERID of the current instance (SYSFJAM) is unknown or the disk space allocated is insufficient to allow creation of the FT-REQUEST-FILE, the file for storing trace data, or the key files.

Response
Either create the CONFIG-USERID or increase its disk space allocation or have your system administrator do it.

FTR1041 `OPENFT: Specified file is not a valid trace file`

FTR1042 `OPENFT: openFT could not be started`

FTR1043 `OPENFT: Partner with same attribute '(&00)' already exists in partner list.`

Meaning
There is already a partner entry with the same attribute '(&00)' in the partner list.

Response
The attribute '(&00)' in partner entries must be unique. Correct the command accordingly and try again.

FTR1044 `OPENFT: Maximum number of partners exceeded.`

Meaning
The partner list already contains the maximum permissible number of partner entries.

Response
Delete partners that are no longer required.

FTR1045 `OPENFT: No partner found in partner list.`

Meaning
A partner for the specified selection could not be found in the partner list.

Response
Check if the specified partner name or address was correct. If necessary, repeat the command using the correct name or address.

FTR1046 `OPENFT: Modification of partner protocol type not possible`

Meaning
The protocol type of the partner entry cannot be changed subsequently.

Response
Delete the partner from the partner list, if necessary, and enter it again with a new protocol type.

FTR1047 `OPENFT: Request (&00) not found.`

Meaning
The request with the transfer ID (&00) could not be found.

Response
Specify the existing transfer ID and repeat the command.

FTR1048 `OPENFT: Active requests could not yet be deleted`

Meaning
Active requests for the specified partner were cancelled. After the negotiation of termination with the partner the requests will be automatically deleted.

FTR1049 `OPENFT: CCS name (&00) unknown`

FTR1057 `OPENFT: Inbound requests cannot be modified`

FTR1059 `OPENFT: Monitoring is not active`

Meaning
The command is only supported if monitoring is activated.

Response
Activate monitoring in the operating parameters.

FTR2014 `OPENFT: No file attribute changes requested.`

Meaning
No further file attributes besides the file name were specified.

Response
Enter the desired file attributes in addition to the file name.

FTR2015 `OPENFT: openFT is not authorized to execute requests for this user`

FTR2016 `OPENFT: Directory (&00) is not empty`

FTR2017 `OPENFT: File attributes do not match request parameters (&00)`

Meaning
The specified attribute combination is not permissible.

Response
Specify a permissible combination.

FTR2018 `OPENFT: Attributes could not be modified (&00).`

Meaning
The properties of the file could not be changed as specified in the command.
The following reasons are possible:
For the remote file:

- No access rights to the file.
- The required combination of access rights is not supported by the remote system.
- If the remote system is a BS2000: the file is protected by ACL.

For the local file:

- No access rights to the file.
- The requested transfer attributes are not compatible with the properties of the file (see manual).

FTR2019 `OPENFT: (&00)' could not be created (&01).`

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receiving system's TRANSFER-ADMISSION to the ID of the receive file's owner. Repeat the command.

FTR2021 `OPENFT: CCS name unknown.`

Meaning
The request could not be completed because the CCS name specified for the local file does not correspond to any of the supported code tables.

FTR2022 `OPENFT: Higher-level directory not found`

Meaning
In the case of a receive request, the local file could not be created because the specified path does not exist.

Response
Create or correct the path for the receive file and repeat the command.

FTR2023 `OPENFT: (&00)' already exists.`

Meaning
The command was not executed because an existing receive file cannot be created again with WRITE-MODE=NEW. WRITE-MODE=NEW may also have been set due to a restriction in the access authorization used.

Response
Either delete the receive file and repeat the command, or repeat the command specifying WRITE-MODE=REPLACE-FILE or using different access authorization.

FTR2024 `OPENFT: Transfer of file generation groups not supported.`

Meaning
The command was not executed because the FT system only transfers single file generations.

Response
Repeat the command using the name of a single file generation.

FTR2025 `OPENFT: Error accessing '(&00)'(&02).`

Meaning
(&02): Further details, possibly DMS error
The FT system continues to run after the message has been issued.

Response
Take the appropriate action in accordance with the error code.

FTR2026 `OPENFT: Resulting file name '(&00)' too long (&01).`

Meaning
The relative file name was specified in the transfer request. The absolute file name completed by openFT is longer than permitted.

Response
Shorten the file name or path and repeat the command.

FTR2027 `OPENFT: No file or directory name specified.`

Meaning
The command was not executed because the file name was neither specified explicitly nor by the 'TRANSFER-ADMISSION' used.

Response
Repeat the command, specifying the file ID explicitly or a TRANSFER-ADMISSION that defines the file ID.

FTR2028 `OPENFT: Invalid management password.`

FTR2029 `OPENFT: (&00)' not available (&01).`

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk, or an attempt has been made to transfer a file migrated by HSM without specifying the local transfer admission (TRANSFER-ADMISSION operand).

Response
Inform the operator if necessary or carry out an HSM recall for the file or specify the local transfer admission. Repeat the command.

FTR2030 `OPENFT: Home directory not found (&00)`

FTR2031 `OPENFT: Renaming not possible (&00)`

FTR2032 `OPENFT: Not enough space for (&00).`

Meaning
The command was not (fully) executed because the permissible storage space on the receive system is used up for the user ID specified in TRANSFER-ADMISSION.
The receive file can not be created/extended after the problem occurs.

Response
Take the appropriate action depending on the cause of the error:

- delete all files no longer required on the receive system, or
- ask the system administrator to allocate more storage space, or
- increase the receive file's primary/secondary allocation.

If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2033 `OPENFT: File owner unknown.`

Meaning
The command was not executed because the owner of either the send file or the receive file was not defined in the local system or because the file owner and the user requesting the creation of a receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2034 `OPENFT: Invalid file password.`

Meaning
The command was not executed because the password for the send file or the receive file is missing or incorrect.

Response
Correct the password in the file description or the command.
Repeat the command.

FTR2036 `OPENFT: Retention period of file not yet expired.`

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired (RETENTION PERIOD).

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2037 `OPENFT: '(&00)' is read only.`

FTR2038 `OPENFT: File structure not supported (&00).`

FTR2039 `OPENFT: Syntax error in resulting file name '(&00)' (&01).`

Meaning
The local file cannot be accessed because, for example, the absolute file name is too long.

Response
Shorten the path or file name. Repeat the command.

FTR2040 `OPENFT: Transparent file transfer not supported.`

Meaning
The request could not be carried out because the partner system does not support the receipt of files in a transparent format.

FTR2042 `OPENFT: Extension of file not possible for transparent transfer.`

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

Response
Start transfer without EXTEND.

FTR2043 `OPENFT: Access to '(&00)' denied (&01).`

Meaning
The command was not executed because either the send file or the receive file only permits certain access modes (e.g. read only).

Response
Correct the file name or file protection attributes. Repeat the command.

FTR2044 `OPENFT: Follow-up processing exceeds length limit.`

Meaning
Prefix + suffix (from prof) + local follow-up processing together are too long.

Response
Correct the file name or file protection attributes. Repeat the command.

FTR2045 `OPENFT: Processing admission invalid.`

Meaning
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands were incorrect.

Response
Define the required PROCESSING ADMISSION or correct it.
Repeat the command if necessary.

FTR2046 `OPENFT: Local transfer admission invalid.`

Meaning
The command was not executed because the specifications in one of the TRANSFER-ADMISSION operands were incorrect.

Response
Define the required TRANSFER ADMISSION or correct it.
Repeat the command if necessary.

FTR2047 `OPENFT: Request rejected by local FTAC.`

Meaning
The command was not executed because the request was rejected by the product openFT-AC due to a lack of authorization.

Response
Use the return code in the logging record to determine and remove the cause.
Repeat the command.

FTR2048 `OPENFT: Function not supported for protocol '(&00)'.`

Meaning
The desired function is not available for the selected protocol.

Response
Select a different protocol.

FTR2049 `OPENFT: Remote follow-up processing not supported`

Meaning
Remote follow-up processing is only available for the openFT protocol.

Response
Select a different protocol, or specify follow-up processing by means of an FTAC profile.

FTR2050 `OPENFT: Data integrity check not supported.`

Meaning
The partner system does not support the data integrity check function.

Response
Repeat the request without a file integrity check.

FTR2051 `OPENFT: User data encryption not possible for this request.`

Meaning
The partner system does not support the data encryption function.

Response
Repeat the request without data encryption or install openFT-CR (or have it installed) on the remote system.

FTR2070 `OPENFT: Request (&00). openFT is no longer authorized to execute requests for this user`

FTR2071 `OPENFT: Request (&00). User data encryption not installed.`

Meaning
The user data encryption function cannot be used unless openFT-CR is installed.

Response
Use openFT-CR.

FTR2072 `OPENFT: Request (&00) has been canceled.`

Meaning
The FT request was canceled because
- the command NCANCEL was specified, or
- the time specified in NCOPY has been reached.

Follow-up processing has been started for the local system, provided no error occurred. Follow-up processing is started for the remote system once all the resources are allocated. Local errors are indicated by the message FTR0604 at the start of follow-up processing.

FTR2073 `OPENFT: Request (&00). Encryption error.`

FTR2074 `OPENFT: Request (&00). '(&01)' could not be created (&02).`

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2075 `OPENFT: Request (&00). Higher-level directory no longer found`

FTR2076 `OPENFT: Request (&00). I/O error for '(&01)'(&02).`

Meaning
The file can no longer be accessed. It may have been deleted during a transfer.

Response
Repeat the request.

FTR2077 `OPENFT: Request (&00). File now locked to prevent multiple access.`

Meaning
The command was not executed because the send file or the receive file is already locked by another process so that it cannot be simultaneously updated.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files that are not closed correctly. If the lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2078 `OPENFT: Request (&00). '(&01)' no longer available (&02).`

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk.

Response
Inform the operator if necessary.
Repeat the command.

FTR2079 `OPENFT: Request (&00). '(&01)' no longer found.`

Meaning
The local send or receive file can no longer be accessed because, for example, it was deleted during an interruption of the openFT system.

Response
Restore the file.
Repeat the command.

FTR2080 `OPENFT: Request (&00). Home directory no longer found (&01)`

FTR2081 `OPENFT: Request (&00). '(&01)' gets no more space.`

Meaning
The command was not executed (any further) executed because
- the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up, or
- the receive file has already reached the maximum number of allocations.

Take the appropriate action depending on the cause of the error:

Response
delete all files no longer required on the receive system, or
- ask the system administrator to allocate more storage space, or
- remove empty blocks from the send file, or
- reorganize the file so that it requires fewer allocations, or
- increase the receive file's primary/secondary allocation.

If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2082 `OPENFT: Request (&00). File owner no longer known.`

Meaning
The command was not executed because the owner of the send file or receive file is not defined on the relevant system or because the file owner and the user who wants to create a receive file are not the same.

Response
Define the file owner, or correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2083 `OPENFT: Request (&00). Pre-/post-processing error(&01).`

Meaning
The command executed as part of local pre-/post-processing returned a result other than OK.

Response
Correct and repeat the command.

FTR2084 `OPENFT: Request (&00). Exit code (&01) for pre-/post-processing (&02).`

Meaning
The command executed as part of local pre-/post-processing returned the exit code (&01).

Response
Correct the command using the exit code (&00) and issue it again.

FTR2085 `OPENFT: Request (&00). File password no longer valid.`

Meaning
The command was not executed because the password for send file or the receive file is missing or incorrect.

Response
Correct the password in the file description or the command.
Repeat the command.

FTR2086 `OPENFT: Request (&00). '(&01)' is now read only.`

FTR2087 `OPENFT: Request (&00). File structure error(&01).`

Meaning
The command was executed due to a file structure error.
File structure errors include:

- The attributes of the send file are incomplete.
- The data of the send file is incompatible with its structure attributes.
- The records of the send file are too long.
- If WRITE-MODE=EXTEND-FILE or -e is specified, the send file and receive file have different structures (e.g. fixed-/variable-length records).
- The send file or receive file in a remote BS2000 system is a member of an old LMS library (not PLAM).

Response
Correct the file or file attributes. If WRITE-MODE=EXTEND-FILE or -e is specified, restore the receive file. Repeat the command.

FTR2088 `OPENFT: Request (&00). NDMS error (&01).`

Meaning
The request was rejected because the partner system currently does not have the resources available to accept requests.

Response
Repeat the request a little later.

FTR2089 `OPENFT: Request (&00). Recovery failed (&01).`

Meaning
The restart attempts were unsuccessful (for example, a pre-/post-processing command could not be completed before the termination of openFT).

Response
Repeat the command.

FTR2090 `OPENFT: Request (&00). Error in file transfer completion.`

Meaning
An error occurred during the final phase of the file transfer. If it was a long transfer, the recipient is advised to check if the file has still been transferred correctly. However, error follow-up processing will be started if it was specified.

Response
Repeat the request, if necessary.

FTR2091 `OPENFT: Requests only partially completed; (&00) of (&01) files were transferred`

Meaning
In the case of a synchronous send request with wildcards, not all files were successfully transferred.

Response
Transfer unsuccessfully transferred files again.

FTR2092 `OPENFT: Request (&00). Access to '(&01)' no longer permissible (&02).`

Meaning
The command was not executed because either the send file or the receive file only permits certain access modes (e.g. read only) or because a directory was specified as either the source or destination of a file transfer.

Response
Correct the transfer direction, write mode, file name or file protection attributes.
Repeat the command.

FTR2094 `OPENFT: Request (&00). Retention period of file not yet expired.`

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired (RETENTION PERIOD).

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2095 `OPENFT: Request (&00). Extension of file not possible for transparent transfer.`

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

Response
Start transfer without EXTEND.

FTR2096 `OPENFT: Request (&00). File structure not supported (&01).`

FTR2097 `Request (&00). Resulting file name '(&01)' too long(&02)`

Meaning
The relative file name was specified in the transfer request. The absolute file name as extended by openFT is longer than permitted.

Response
Shorten the file name or path and repeat the command.

FTR2109 `OPENFT: Request (&00). Connection setup rejected by local transport system.`

FTR2110 `OPENFT: Request (&00). Data integrity check indicates an error.`

Meaning
The integrity of the data was violated.

FTR2111 `OPENFT: Encryption/data integrity check not possible. Encryption switched off.`

Meaning
There is no key pair set or the key length was set to 0. Requests can only be carried out without data encryption or a data integrity check.

Response
Repeat the request without data encryption, create a key or set a key length >0.

FTR2112 `OPENFT: Request (&00). Data integrity check not supported by partner.`

Meaning
The partner system does not support the data integrity check.

Response
Repeat the request without a data integrity check.

FTR2113 `OPENFT: Request (&00). User data encryption not possible for this request.`

Meaning
The partner system does not support the data encryption function.

Response
Repeat the request without data encryption or install openFT-CR (or have it installed) on the remote system.

FTR2114 `OPENFT: Request (&00). Identification of local system rejected by remote system '(&01)'.`

Meaning
For security reasons or because of an inconsistency, the partner did not accept the instance identification of the local system (for example, because in a network description file both the instance identification and migration identification %.prozessor.entity occur for different partners).

Response
Ensure that the local identification has been entered correctly on the partner system and has not been assigned to a different partner.

FTR2115 `OPENFT: Request (&00). Interrupted by remote system`

FTR2116 `OPENFT: Local application (&00) not defined`

Meaning
The local application is not defined in the transport system, or the tnsxd process will not run in the Unix system.

Response
Make the local application known to the local transport system or start the tnsxd process.

FTR2117 `OPENFT: Local application (&00) not available`

FTR2118 `OPENFT: Request (&00). Authentication of local system failed.`

Meaning
The local system could not be authenticated by the partner system.

Response
Give the current public key file to the partner and name it correctly there. Repeat the command.

FTR2119 `OPENFT: Request (&00). Local system unknown in remote system.`

Meaning
The local system is not known on the partner system (e.g. BS2000/OSD or z/OS).

Response
Make the local system known on the partner system and repeat the command.

FTR2120 `OPENFT: Remote system '(&00)' unknown.`

Meaning
The partner specified as the remote system cannot be expanded to an address on the local system.

Response
Correct the specification for the partner or add the partner to the partner list and repeat the command.

FTR2121 `OPENFT: Request (&00). Authentication of partner failed.`

Meaning
The remote system could not be authenticated by the local system.

Response
Get the current public key file from the partner and name it correctly.

FTR2122 `OPENFT: Request (&00). FT session rejected or disconnected. Reason (&01)`

FTR2123 `OPENFT: Request (&00). OSS call error (&01).`

Meaning
The command was not executed because the session instance detected a communication error.
(&00): error code.

Response
Take the appropriate action in accordance with the error code.

FTR2124 `OPENFT: Request (&00). No free connection`

Meaning
No more transfers are possible because the maximum number of simultaneous transfers has been reached.

Response
Check whether the transport system is working (or have it checked).

FTR2125 `OPENFT: Request (&00). Connection lost.`

Meaning
No data transfer took place because of a line interrupt or a line protocol error.

Response
Repeat the request.

FTR2126 `OPENFT: Request (&00). Transport system error. Error code (&01)`

Meaning
An error occurred in the transport system during processing of a FTSTART command or a file transfer or file management request.

Response
Take the appropriate action in accordance with the error code. Most often the occurrence of this message indicates that the partner addressed is not known to the transport system.

FTR2127 `OPENFT: Request (&00). No data traffic within (&01) seconds`

Meaning
No data transfer took place within the period of seconds specified because, for example, the connection is interrupted, the partner is not sending and the local system is waiting for data.

Response
Repeat the request.

FTR2128 `OPENFT: OSS version not supported`

Meaning
At least OSS version V04.1 required.

FTR2140 `OPENFT: Request (&00). Remote system: openFT is not authorized to execute requests for this user.`

FTR2141 `OPENFT: Request (&00). Remote system: Directory (&01) is not empty`

Meaning
The command could not be executed because there are files in the specified directory of the partner system.

Response
Delete all the files in the directory first and repeat the command.

FTR2142 `OPENFT: Request (&00). Remote system: File attributes do not match the request parameters (&01)`

Meaning
The command could not be executed because the file attributes on the remote system do not agree with the request parameters (e.g. a directory was specified instead of a remote file).

Response
Check the file name on the remote system and correct it. Repeat the command.

FTR2143 `OPENFT: Request (&00). Remote system: Attributes could not be modified (&01).`

Meaning
The properties of the file could not be modified as desired in the command.
Possible reasons are for the remote file:

- No access rights to the file.
- The combination of access rights required is not supported by the remote system.
- If the remote system is a BS2000: the file is protected by ACL.

FTR2144 `OPENFT: Request (&00). Remote system: File/directory (&01) could not be created (&02)`

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER-ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2145 `OPENFT: Request (&00). Remote system: CCS name unknown or not supported.`

Meaning
The request could not be completed because the CCS is unknown in the partner system.

FTR2146 `OPENFT: Request (&00). Remote system: Higher-level directory not found`

Meaning
The command was not executed because the higher-level directory could not be found on the partner system.

Response
Create the directory on the remote system or correct the remote directory name and repeat the command.

FTR2147 `OPENFT: Request (&00). Remote system: File/directory '(&01)' already exists.`

Meaning
The command was not executed. Possible reasons:

- The command was not executed because an existing receive file cannot be created with 'WRITE-MODE=NEW' or the -n option. WRITE-MODE=NEW or -n may also have been set by a restriction in the access authorization used.
- ftcredir: The specified directory already exists.

Response
Either delete the receive file before repeating the command or reenter the command specifying WRITE-MODE=REPLACE-FILE or using different access authorization.

FTR2148 `OPENFT: Request (&00). Remote system: Transfer of file generation groups not supported.`

Meaning
The command was not executed because the FT system can only transfer single file generations.

Response
Repeat the command using the name of a single file generation.

FTR2149 `OPENFT: Request (&00). Remote system: Access error for '(&01)' (&02).`

Meaning
(&02): DMS error, possibly the transfer ID. The FT system continues to run after output of the message.

Response
Take the appropriate action in accordance with the error code.

FTR2150 `OPENFT: Request (&00). Remote system: Resulting file name too long (&01).`

Meaning
A syntax error other than 'operand missing' (FTR0010) or 'keyword unknown' (FTR0011) has been detected. Possible reasons:

- Values assigned outside the valid range
- Invalid operand separators
- Invalid value assignment characters
- Partially qualified file names

Response
Repeat the command using the correct syntax.

FTR2151 `OPENFT: Request (&00). Remote system: File locked to prevent multiple access.`

Meaning
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously.

Response
Repeat the command later or unlock the file on the remote system. After a system crash in BS2000 you may need to verify files not closed correctly. If the lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2152 `OPENFT: Request (&00). Remote system: No file or directory name specified.`

Meaning
The command was not executed because the file ID was neither specified explicitly nor by Repeat the command, specifying the file ID explicitly or using a TRANSFER ADMISSION that defines the file ID.

FTR2153 `OPENFT: Request (&00). Remote system: Invalid management password.`

FTR2154 `OPENFT: Request (&00). Remote system: File/directory '(&01)' not available (&02).`

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, the file extends over more than one private disk, or an attempt has been made to transfer a file migrated by HSM without specifying the remote transfer admission.

Response
Inform the operator if necessary or carry out an HSM recall for the file or specify the remote transfer admission. Repeat the command.

FTR2155 `OPENFT: Request (&00). Remote system: File/directory '(&01)' not found.`

Meaning
The command was not executed because the send file is not or no longer in the catalog or on a volume of the remote system.

Response
Correct the remote file name, read the file in from tape or restore the send file.
Repeat the command.

FTR2156 `OPENFT: Request (&00). Remote system: Home directory not found (&01)`

FTR2157 `OPENFT: Request (&00). Remote system: Renaming not possible (&01)`

FTR2158 `OPENFT: Request (&00). Remote system: Not enough space for '(&01).`

Meaning
The command was not executed (any further) because the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up. The receive file is no longer created/extended after the problem has occurred.

Response
Take the appropriate action depending on the cause of the error:
- delete all files no longer required on the receive system, or
- ask the system administrator to allocate more storage space, or
- increase the receive file's primary/secondary allocation.

If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2159 `OPENFT: Request (&00). Remote system: File owner unknown.`

Meaning
The command was not executed because the owner of either the send file or the receive file was not defined on the relevant system or because the file owner and the user requesting the creation of a receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2160 `OPENFT: Request (&00). Remote system: Invalid file password.`

Meaning
The command was not executed because the password for the send file or the receive file is missing or incorrect.

Meaning
Correct the password in the file description or the command. Repeat the command.

FTR2161 `OPENFT: Request (&00). Remote system: Retention period of file not yet expired.`

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired.

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2162 `OPENFT: Request (&00). Remote system: File/directory '(&01)' is read only.`

Meaning
The file or directory is write-protected.

Response
Correct the remote file name or remove the write protection of the remote file.
Repeat the command.

FTR2163 `OPENFT: Request (&00). Remote system: File structure not supported(&01).`

Meaning
The request cannot be carried out because the file structure is not supported. For example, an attempt was made to get a PLAM library or ISAM file from the BS2000 system.

Response
Transfer the file transparently.

FTR2164 `OPENFT: Request (&00). Remote system: Syntax error in resulting file name(&01).`

Meaning
A syntax error other than 'operand missing' (FTR0010) or 'keyword unknown' (FTR0011) has been detected.
Possible reasons:

– Values assigned outside the valid range
– Invalid operand separators
– Invalid value assignment characters
– Partially qualified file names

Meaning
Repeat the command using the correct syntax.

FTR2165 `OPENFT: Request (&00). Remote system: Transparent file transfer not supported.`

Meaning
The request could not be carried out because the partner system does not support the transfer of files in a transparent format.

FTR2166 `OPENFT: Request (&00). Remote system: Extension of file not possible for transparent transfer.`

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

FTR2167 `OPENFT: Request (&00). Remote system: Access to '(&01)' denied (&02).`

Meaning
The command was not executed because the remote file only permits certain access modes.

Response
Correct the transfer direction, file name or file protection attributes on the remote system. Repeat the command.

FTR2168 `OPENFT: Request (&00). Remote system: Follow-up processing exceeds length limit.`

Meaning
The length of follow-up processing was exceeded; see the command syntax description.

Response
Shorten the follow-up processing, or use procedures. Repeat the command.

FTR2169 `OPENFT: Request (&00). Remote system: Transfer admission invalid.`

Meaning
The command was not executed because the specifications in one of the TRANSFER-ADMISSION operands are incorrect or the request was rejected by FTAC because of insufficient authorization.

Response
Define the requisite TRANSFER-ADMISSION or correct it or check the authorization entered in FTAC. Repeat the command if necessary.

FTR2170 `OPENFT: Request (&00). Remote system: Function not supported (&01).`

FTR2171 `OPENFT: Request (&00). Remote system: Processing admission invalid.`

Meaning
The command was not executed because the specifications in one of the PROCESSING-ADMISSION operands are incorrect.

Response
Define the required PROCESSING ADMISSION or correct it.
Repeat the command if necessary..

FTR2172 `OPENFT: Request (&00). Remote system: Request queue full.`

Meaning
The command was not executed because the maximum number of permissible file transfer requests has been reached.

Response
Notify the FT administrator. Repeat the command later.

FTR2195 `OPENFT: Request (&00). Remote system: openFT is not longer authorized to execute requests for this user.`

FTR2196 `OPENFT: Request (&00) has been canceled in the remote system.`

Meaning
The request was deleted on the remote system before termination.

FTR2197 `OPENFT: Request (&00). Remote system: File/directory '(&01)' could not be created(&02).`

Meaning
The command was not executed because the file owner and user requesting the creation of a receive file are not the same.

Response
Match the user ID in the receive system's TRANSFER-ADMISSION to the ID of the receive file owner. Repeat the command.

FTR2198 `OPENFT: Request (&00). Remote system: Higher-level directory no longer found`

FTR2199 `OPENFT: Request (&00). Remote system: I/O error for '(&01)' (&02).`

Meaning
An error occurred at input/output. Possible cause:
- BS2000: DMS error, possibly the transfer ID.
- The send or receive files was deleted during transfer.

The FT system continues to run after the message has been issued.

Response
Take the appropriate action in accordance with the error code.

FTR2200 `OPENFT: Request (&00). Remote system: File now locked to prevent multiple access.`

Meaning
The command was not executed because either the send file or the receive file is already locked by another process to prevent it from being updated simultaneously. An attempt is made, for example, to access a library opened in z/OS.

Response
Repeat the command later or unlock the file. After a system crash you may need to verify files not closed correctly. If a lock is caused by an FT request, it will be released automatically when the request is finished.

FTR2201 `OPENFT: Request (&00). Remote system: File/directory '(&01)' no longer available(&02).`

Meaning
The command was not executed because the volume for either the send file or the receive file is not mounted, unknown or reserved, or because the file extends over more than one private disk or an attempt has been to transfer a file migrated by HSM.

Response
Inform the operator if necessary or carry out an HSM recall for the file.
Repeat the command.

FTR2202 `OPENFT: Request (&00). Remote system: File/directory '(&01)' no longer found.`

Meaning
The command was not executed because the remote file is not or no longer in the catalog or on a volume of the corresponding system (e.g. after a restart).

Response
Restore the remote file. Repeat the command.

FTR2203 `OPENFT: Request (&00). Remote system: Home directory no longer found (&01)`

FTR2204 `OPENFT: Request (&00). Remote system: File/directory '(&01)' gets no more space.`

Meaning
The command was not executed (any further) because

- the permissible storage space on the receive system for the user ID specified in TRANSFER-ADMISSION has been used up, or
- the send file contains too long a sequence of empty blocks, or
- the primary and/or secondary allocation of the password-protected receive file is too small.

The receive file can no longer be created/extended after the problem occurs.

Response
Take the appropriate action depending on the cause of the error:

- delete all files no longer required on the receive system, or
- ask the system administrator to allocate more storage space, or
- remove empty blocks from the send file, or
- increase the receive file's primary/secondary allocation.

If WRITE-MODE=EXTEND-FILE is specified, restore the receive file.
Repeat the command.

FTR2205 `OPENFT: Request (&00). Remote system: File owner no longer known.`

Meaning
The command was not executed because the owner of either the send file or the receive file is not defined on the relevant system, or because the file owner and the user requesting the creation of the receive file are not the same.

Response
Define the file owner, correct TRANSFER-ADMISSION or FILE-NAME.
Repeat the command.

FTR2206 `OPENFT: Request (&00). Remote system: Pre-/post-processing error (&01).`

Meaning
The command executed in local pre-/postprocessing returned a result value other than OK.

Response
Correct the pre-/post-processing command and issue it again.

FTR2207 `OPENFT: Request (&00). Remote system: Exit code (&01) during pre-/post-processing (&02).`

Meaning
The command executed in local pre-/postprocessing returned the exit code (&01).

Response
Correct the pre-/post-processing command in accordance with the exit code and issue it again.

FTR2208 `OPENFT: Request (&00). Remote system: File password no longer valid.`

Meaning
The command was not executed because the password for the send file or receive file is missing or incorrect.

Response
Correct the password in the file description or the command. Repeat the command.

FTR2209 `OPENFT: Request (&00). Remote system: File/directory '(&01)' is now read only.`

FTR2210 `OPENFT: Request (&00). Remote system: File structure error (&01).`

Meaning
The command was not executed due to a file structure error.
File structure errors include:

- The attributes of the send file are incomplete.
- The data of the send file is incompatible with its structure attributes.
- The records of the send file are too long.
- If WRITE-MODE=EXTEND-FILE or the -e parameter are specified, the send file and receive file have different structures (e.g. fixed-/variable-length records).
- BS2000: The send or receive file is a member of an old LMS library (not PLAM).
- BS2000: The send file has an odd block factor (e.g. BLKSIZE=(STD,1)), and the receive file is stored on an NK4 pubset.

Response
Correct the file or file attributes. If WRITE-MODE=EXTEND-FILE is specified, restore the receive file. Repeat the command.

FTR2211 `OPENFT: Request (&00). Remote system: NDMS error (&01).`

Response
Repeat the request a little later.

FTR2212 `OPENFT: Request (&00). Recovery failed (&01).`

Meaning
The restart could not be carried out. It may not have been possible to complete restart-capable pre-/post-processing before termination of the server process (waiting time: max. minutes).

Response
Repeat the command.

FTR2213 `OPENFT: Request (&00). Remote system: Resource bottleneck.`

Meaning
The order was rejected because the partner system currently does not have the resources available to accept requests. It is possible that the maximum number of concurrent connections in the partner system does not permit any additional connection.

Response
Repeat the request a little later. Where necessary, ask the administrator of the partner system to increase the maximum number of concurrent connections on their system.

FTR2214 `OPENFT: Request (&00). Remote system: Access to '(&01)' is no longer permissible(&02).`

Meaning
The command was not executed because

– the send file or receive file only permits certain access modes (e.g. read only) or a directory was specified as the source or destination of a file transfer.
– or because no valid password for an FTAC profile has been stored in the local system for executing the ftexec command from a remote system.

Response
Correct the transfer direction, write mode, file name or file protection attributes or specify a valid password for the FTAC profile. Repeat the command.

FTR2216 `OPENFT: Request (&00). Remote system: File structure not supported (&01).`

Meaning
The request cannot be carried out because the file structure is not supported. An attempt was made, for example, to get a PLAM library or ISAM file from BS2000.

Response
Transfer the file transparently.

FTR2217 `OPENFT: Request (&00). Remote system: Retention period of file not yet expired.`

Meaning
The command was not executed because the retention period protecting the receive file against overwriting has not yet expired.

Response
Correct the transfer direction, retention period or file name. Repeat the command.

FTR2218 `OPENFT: Request (&00). Remote system: Extension of file not possible for transparent transfer.`

Meaning
The command could not be executed because it is not possible to add to a file in a transparent transfer.

FTR2225 `OPENFT: Information output canceled.`

Meaning
A show command was interrupted, for example.

Response
Repeat the command.

9.9.3 FTC messages

FTC0001 FTAC VERSION (&00) ACTIVE

Meaning
FTAC initialization is concluded.

FTC0003 (&00) LOGGING RECORDS DELETED

Meaning
The specified number of records have been deleted from the logging file.

FTC0050 CMD ACCEPTED. WARNING: LOWER ADM-LEVEL REMAINS IN EFFECT

Meaning
The set security level exceeds the administrator’s limit value and will remain without effect until the administrator’s limit value is increased.

Response
Request a higher maximum security level from the FTAC administrator.

FTC0051 CMD ACCEPTED. WARNING: TRANSFER-ADMISSION EXISTS AS USER ID

Meaning
A user ID with the same name already exists in the system.

Response
The message is simply intended to indicate a possible confusion.

FTC0052 CMD TERMINATED. INFORMATION INCOMPLETE

Meaning
Information output has been interrupted.

Response
Repeat the command if necessary.

FTC0053 CMD TERMINATED. NO FT PROFILE FOUND

Meaning
There is no FT profile for the specified criteria.

FTC0054 CMD ACCEPTED. NO INFORMATION AVAILABLE

Meaning
There is no information on the specified criteria.

FTC0055 WARNING: PARTNER RESTRICTION DOES NOT LONGER EXIST

FTC0056 WARNING: TRANSFER ADMISSION LOCKED

FTC0057 `WARNING: ATTRIBUTES OF TRANSFER ADMISSION ARE IGNORED`

Meaning
In the case of a profile with transfer admission *NOT-SPECIFIED, VALID, USAGE and EXPIRATION-DATE are ignored.

FTC0070 `CMD TERMINATED. SHORTAGE OF RESOURCES`

Meaning
The command cannot be executed due to a lack of resources.

Response
Repeat the command.

FTC0071 `CMD REJECTED. OPENFT NOT ACTIVE`

Meaning
FT has not been activated, FTAC is therefore inactive.

Response
Ask the system administrator to activate FT. FTAC will be activated by FT.

FTC0100 `CMD REJECTED. FT PROFILE ALREADY EXISTS`

Meaning
There is already an FT profile with the specified name.

Response
Select another name.

FTC0101 `CMD REJECTED. TRANSFER ADMISSION ALREADY EXISTS`

Meaning
There is already an FT profile with the specified transfer admission.

Response
You should choose the TRANSFER-ADMISSION more carefully to ensure greater security.

FTC0102 `FILE ALREADY EXISTS`

FTC0103 `INVALID FILE CONTENT OR ACCESS TO FILE DENIED`

Meaning
The file is not an FTAC export file or access is prohibited.

FTC0104 `ACCESS TO USER ID DENIED OR USER ID DOES NOT EXIST.`

Meaning
Access to user ID rejected.
The user ID does not exist.

FTC0105 `ACCESS TO FILE DENIED`

FTC0106 `ACCESS TO TEMPORARY FILE DENIED`

FTC0107 `NO SPACE AVAILABLE`

FTC0108 `THE VERSION OF EXPORT FILE IS NOT COMPATIBLE WITH CURRENT VERSION`

FTC0109 `FILE IS NO FTAC EXPORT FILE`

FTC0110 `FILE NAME TOO LONG`

FTC0111 `SYNTAX ERROR IN FILE NAME`

FTC0112 `CMD REJECTED. EXPIRATION DATE NOT VALID`

Meaning
The value of the parameter EXPIRATION-DATE must be between 1970-01-02 and 2038-01-19.

FTC0150 `CMD REJECTED. USER NOT AUTHORIZED FOR FTAC COMMANDS`

Meaning
There is no password for the admission.

Response
Specify the FTAC password.

FTC0151 `CMD REJECTED. USER NOT AUTHORIZED FOR THIS MODIFICATION`

Meaning
Only the administrator or owner can perform the modification.

FTC0152 `CMD REJECTED. USER NOT AUTHORIZED FOR OTHER USER IDS`

Meaning
The specified user ID is not your own user ID.

FTC0153 `CMD REJECTED. USER NOT AUTHORIZED FOR OTHER OWNER IDS`

Meaning
The specified owner identification is not your own user ID.

FTC0154 `CMD REJECTED. NO AUTHORIZATION FOR DELETION OF LOG RECORDS`

FTC0155 `CMD REJECTED. USER NOT AUTHORIZED FOR DIAGNOSE`

Meaning
Only the FT administrator and FTAC administrator may call the diagnostic function.

FTC0156 `COMMAND ALLOWED FOR FTAC ADMINISTRATOR ONLY`

FTC0157 `CMD REJECTED. NO AUTHORIZATION FOR THIS SET OF PARAMETERS`

Meaning
The FTAC administrator can only create profiles with a transfer admission specification if he or she knows the complete user ID or possesses the SU privilege.

Response
Specify the full user ID in the form user-adm=(uid,acc,pw).

FTC0170 `CMD REJECTED. GIVEN PARTNER UNKNOWN`

Meaning
The specified partner is unknown within the group of partner systems permitted for this user.

FTC0171 `CMD REJECTED. GIVEN FT PROFILE NAME UNKNOWN`

Meaning
The specified profile does not exist.

FTC0172 `CMD REJECTED. INVALID USER ADMISSION`

Meaning
The specified user admission does not exist in the system.

Response
The USER-IDENTIFICATION, ACCOUNT or PASSWORD is incorrect.

FTC0173 `CMD REJECTED. INVALID PROCESSING ADMISSION`

Meaning
The specified processing admission does not exist in the system.

Response
The USER-IDENTIFICATION, ACCOUNT or PASSWORD specification is incorrect.

FTC0174 `CMD REJECTED. MODIFICATION INVALID FOR NOT UNIQUE SELECTION CRITERIA`

Meaning
The parameters "NEW-NAME" and "TRANSFER-ADMISSION" may only be used in combination with unique selection criteria ("NAME" or "TRANSFER-ADMISSION").

Response
Choose a unique selection criterion.

FTC0175 `CMD REJECTED. MODIFICATION INVALID FOR STANDARD AUTHORIZATION RECORD`

Meaning
The parameter "NEW-PASSWORD" may not be specified for *STD.

FTC0176 `CMD REJECTED. GIVEN USER ID UNKNOWN`

Meaning
The specified user ID does not exist in the system.

FTC0177 `FILE UNKNOWN`

FTC0178 `MULTIPLE PARTNER NAME SPECIFIED`

FTC0179 `VIOLATION OF MAXIMAL NUMBER OF PARTNER RESTRICTIONS`

FTC0180 `MULTIPLE USERID SPECIFIED`

FTC0181 `MULTIPLE FT PROFILE NAME SPECIFIED`

FTC0182 `TOTAL MAXIMUM PARTNER NAME LENGTH EXCEEDED`

Meaning
The total length of the partner names may not exceed 1000 characters.

FTC0183 `CMD REJECTED. PARTNER NOT SUPPORTED`

FTC0184 `Invalid parameter transfer admission for profile *STD`

Meaning
The transfer admission of the default profile must be *NOT-SPECIFIED.

FTC0185 `COMBINATION OF THESE TRANSFER FUNCTIONS NOT ALLOWED`

FTC0200 `CMD REJECTED. FOLLOW-UP PROCESSING TOO LONG`

Meaning
The total length of the two follow-up processing commands is too great.

Response
Use shorter commands (e.g. by using procedures).

FTC0201 `USER ID TOO LONG`

FTC0202 `PROFILE NAME TOO LONG`

FTC0203 `TRANSFER ADMISSION TOO LONG`

FTC0204 `PARTNER TOO LONG`

FTC0205 `FULLY QUALIFIED FILE NAME TOO LONG`

FTC0206 `PARTIALLY QUALIFIED FILE NAME TOO LONG`

FTC0207 `PROCESSING COMMAND TOO LONG`

FTC0208 `INVALID DATE SPECIFIED`

FTC0209 `INVALID TIME SPECIFIED`

FTC0210 `TRANSFER ADMISSION TOO SHORT`

FTC0211 `PARAMETERS (&00) AND (&01) MAY NOT BE SPECIFIED TOGETHER`

FTC0212 `LICENSE CHECK ERROR (&00) FOR FTAC`

FTC0213 `MANDATORY PARAMETER PROFILE NAME IS MISSING`

FTC0214 `MANDATORY PARAMETER FILE NAME IS MISSING`

FTC0215 `SYNTAX ERROR IN PARAMETER (&00)`

FTC0216 `PASSWORD TOO LONG`

FTC0217 `TEXT TOO LONG`

FTC0218 `TOO MANY PARTNERS`

FTC0219 `TOO MANY USERS`

FTC0220 `TOO MANY PROFILES`

FTC0250 `LOAD ERROR. ERROR-CODE (&00)`

FTC0251 `CMD REJECTED. FTAC NOT AVAILABLE`

Meaning
openFT-AC has not been installed completely.

Response
The system administrator must check the openFT-AC installation.

FTC0253 `FTAC COMMAND NOT FOUND IN SYNTAXFILE`

Meaning
The openFT-AC syntax file has been merged incorrectly or incompletely into the system syntax file.

Response
The system administrator must check the system syntax file.

FTC0254 `SYSTEM ERROR. ERRORCODE (&00)`

Meaning
A system error has occurred.

Response
Generate diagnostic material and inform the staff responsible for system diagnostics.

FTC0255 `CMD TERMINATED. SYSTEM ERROR`

Meaning
A system error has occurred.

Response
Inform the system administrator. At the same time a message is issued to the operator terminal providing exact troubleshooting information.

FTC100A `SUBMISSION REJECTED. VIOLATION OF ELEMENT-TYPE RESTRICTION`

Meaning
The FT profile does not permit the specification ELEMENT-TYPE in the request.

FTC100B SUBMISSION REJECTED. VIOLATION OF FILE-PASSWORD RESTRICTION

Meaning
The FT profile does not permit the specification FILE-PASSWORD in the request.

FTC100C SUBMISSION REJECTED. VIOLATION OF USER-IDENTIFICATION(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification USER-IDENTIFICATION in the request's PROCESSING-ADMISSION.

FTC100D SUBMISSION REJECTED. VIOLATION OF ACCOUNT(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification ACCOUNT in the request's PROCESSING-ADMISSION.

FTC100E SUBMISSION REJECTED. VIOLATION OF PASSWORD(PROCESSING-ADMISSION) RESTRICTION

Meaning
The FT profile does not permit the specification PASSWORD in the request's PROCESSING-ADMISSION.

FTC100F SUBMISSION REJECTED. VIOLATION OF SUCCESS-PROCESSING RESTRICTION

Meaning
The FT profile does not permit the specification SUCCESS-PROCESSING.

FTC1001 SUBMISSION REJECTED. INVALID TRANSFER-ADMISSION

Meaning
The specified TRANSFER-ADMISSION is not defined in any FT profile.

FTC1002 SUBMISSION REJECTED. INVALID INITIATOR

Meaning
The FT profile restricts initiatives to LOCAL or REMOTE.

FTC1003 SUBMISSION REJECTED. INVALID TRANSFER-DIRECTION

Meaning
The FT profile restricts the TRANSFER-DIRECTION to TO or FROM.

FTC1004 SUBMISSION REJECTED. INVALID PARTNER NAME

Meaning
The FT profile does not permit any requests involving the specified partner system.

FTC1005 SUBMISSION REJECTED. VIOLATION OF MAX-PARTNER-LEVEL

Meaning
The partner system's security level exceeds the value specified for MAX-PARTNER-LEVEL in the FT profile.

FTC1006 `SUBMISSION REJECTED. SYNTAX ERROR OF FILE NAME EXPANSION`

Meaning
The FT profile does not permit the specification of a file name or file name expansion in the request.

FTC1007 `SUBMISSION REJECTED. VIOLATION OF LIBRARY RESTRICTION`

Meaning
The file or library name specified in the command infringes the LIBRARY restriction in the profile.

FTC1008 `SUBMISSION REJECTED. VIOLATION OF ELEMENT RESTRICTION`

Meaning
The FT profile does not permit the specification ELEMENT in the request.

FTC1009 `SUBMISSION REJECTED. VIOLATION OF ELEMENT-VERSION RESTRICTION`

Meaning
The FT profile does not permit the specification ELEMENT-VERSION in the request.

FTC1010 `SUBMISSION REJECTED. VIOLATION OF FAILURE-PROCESSING RESTRICTION`

Meaning
The FT profile does not permit the specification FAILURE-PROCESSING.

FTC1011 `SUBMISSION REJECTED. VIOLATION OF WRITE-MODE RESTRICTION`

Meaning
The FT profile does not permit the specified WRITE-MODE.

FTC1012 `SUBMISSION REJECTED. INVALID FT-FUNCTION`

Meaning
The FT profile does not permit the desired FT function.

FTC1013 `SUBMISSION REJECTED. VIOLATION OF PROFILE WITH CHIPCARD-ID`

Meaning
The profile may only be used with a chipcard.

FTC1014 `SUBMISSION REJECTED. VIOLATION OF DATA ENCRYPTION RESTRICTION`

Meaning
The profile does not permit the value DATA-ENCRYPTION in the request.

FTC2001 `SUBMISSION REJECTED. SYNTAX ERROR ON FILE NAME EXPANSION`

Meaning
The combination of the FT profile’s FILE-NAME and FILE-NAME expansion resulted in a syntax error.

FTC2002 `SUBMISSION REJECTED. SYNTAX ERROR ON LIBRARY NAME EXPANSION`

Meaning
The combination of the FT profile’s LIBRARY name and LIBRARY expansion resulted in a syntax error.

FTC2003 `SUBMISSION REJECTED. SYNTAX ERROR ON ELEMENT NAME EXPANSION`

Meaning
The combination of the FT profile’s ELEMENT name and ELEMENT expansion resulted in a syntax error.

FTC2004 `SUBMISSION REJECTED. TOTAL LENGTH OF RESULT PROCESSING EXCEEDS 500 CHARACTERS`

Meaning
SUCCESS and FAILURE processing including the expansions defined in the FT profile exceeds 1000 characters.

FTC3001 `SUBMISSION REJECTED. INVALID USER-IDENTIFICATION`

Meaning
The TRANSFER-ADMISSION’s USER-IDENTIFICATION or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3002 `SUBMISSION REJECTED. INVALID ACCOUNT`

Meaning
The TRANSFER-ADMISSION’s ACCOUNT specification or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3003 `SUBMISSION REJECTED. INVALID PASSWORD`

Meaning
The TRANSFER-ADMISSION’s PASSWORD specification or, if an FT profile is used, the USER-ADMISSION is invalid.

FTC3004 `SUBMISSION REJECTED. TRANSFER ADMISSION LOCKED`

Meaning
The transfer admission is locked. The reasons may be ascertained from the output from the FTSHWPRF command.

FTC3011 `SUBMISSION REJECTED. VIOLATION OF USER OUTBOUND SEND LEVEL`

Meaning
The partner system’s security level is not permitted by the user for the OUTBOUND SEND function class.

FTC3012 `SUBMISSION REJECTED. VIOLATION OF USER OUTBOUND RECEIVE LEVEL`

Meaning
The partner system’s security level is not permitted by the user for the OUTBOUND RECEIVE function class.

FTC3013 SUBMISSION REJECTED. VIOLATION OF USER INBOUND SEND LEVEL

Meaning
The partner system’s security level is not permitted by the user for the INBOUND SEND function class.

FTC3014 SUBMISSION REJECTED. VIOLATION OF USER INBOUND RECEIVE LEVEL

Meaning
The partner system’s security level is not permitted by the user for the INBOUND RECEIVE function class.

FTC3015 SUBMISSION REJECTED. VIOLATION OF USER INBOUND PROCESSING LEVEL

Meaning
The partner system’s security level is not permitted by the user for the INBOUND PROCESSING function class.

FTC3016 SUBMISSION REJECTED. VIOLATION OF USER INBOUND FILE MANAGEMENT LEVEL

Meaning
The partner system’s security level is not permitted by the user for the INBOUND FILE MANAGEMENT function class

FTC3021 SUBMISSION REJECTED. VIOLATION OF ADM OUTBOUND SEND LEVEL

Meaning
The partner system’s security level is not permitted by the administrator for the OUTBOUND SEND function class.

FTC3022 SUBMISSION REJECTED. VIOLATION OF ADM OUTBOUND RECEIVE LEVEL

Meaning
The partner system’s security level is not permitted by the administrator for the OUTBOUND RECEIVE function class.

FTC3023 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND SEND LEVEL

Meaning
The partner system’s security level is not permitted by the administrator for the INBOUND SEND function class.

FTC3024 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND RECEIVE LEVEL

Meaning
The partner system’s security level is not permitted by the administrator for the INBOUND RECEIVE function class.

FTC3025 SUBMISSION REJECTED. VIOLATION OF ADM INBOUND PROCESSING LEVEL

Meaning
The partner system’s security level is not permitted by the administrator for the INBOUND PROCESSING function class.

```
FTC3026    SUBMISSION REJECTED. VIOLATION OF ADM INBOUND FILE MANAGEMENT LEVEL
```

Meaning
The partner system’s security level is not permitted by the administrator for the INBOUND FILE MANAGEMENT function class

9.10 Using openFT in z/OS systems without the TSO interactive system

openFT is intended for use under the z/OS operating system. The commands are passed to the TSO command processor. Nevertheless, openFT can also be used without the TSO interactive system. In this case, the IBM utility IKJEFT01 must be used to call the TSO command processor in batch mode.

In order to be able to work with openFT without the TSO interactive system, all commands must be included in batch jobs. These jobs are initiated via the IBM utility IEBGENER. IEBGENER reads the job information from a file and passes it on to the Job Entry Subsystem (JES2/3).

It is not then possible to set the openFT installation parameters using the FJGEN command (see page 191) within a TSO dialog. Instead, the installation parameters must be set using a parameter library (see section “Setting up the FT parameter library” on page 49).

Issuing TSO commands

These commands are processed by the TSO command processor. In an exclusive z/OS batch environment, the IKJEFT01 utility provides the appropriate interface.

Example of a batch job including the NCOPY command:

```
//USERN      JOB    .....
//NCOPY      EXEC PGM=IKJEFT01
//SYSPRINT   DD   SYSOUT=*
//SYSTSPRT   DD   SYSOUT=*
//SYSTSIN    DD   *
NCOPY TRANS=TO,PARTNER=MVS2,+
LOC=(FILE=......
...
...
/*
//
```

Glossary

Italic type indicates a reference to other terms in this glossary.

ABEND
Abnormal termination of program.

access protection
Comprises all the methods used to protect a data processing system against unauthorized system access.

access right / access admission
Derived from the *transfer admission*. The access right defines the scope of access for the user who specifies the transfer admission.

ACF-2
Program product from Computer Associates for system and data access control.

ADM administrator
Administrator of the *remote administration server*. This is the only person permitted to modify the configuration data of the remote administration server.

ADM partner
Partner system of an openFT instance with which communication takes place over the *FTADM protocol* in order to perform *remote administration*.

ADM traps
Short messages sent to the *ADM trap server* if certain events occur during operation of openFT.

ADM trap server
Server that receives and permanently stores the *ADM traps*. It must be configured as a *remote administration server*.

administrated openFT instance
openFT instances that are able to be administered by *remote administrators* during live operation.

admission profile

Way of defining the *FTAC* protection functions. Admission profiles define a *transfer admission* that has to be specified in *FT requests* instead of the *LOGON or Login authorization*. The admission profile defines the *access rights* for a user ID by restricting the use of parameters in *FT requests*.

admission profile, privileged

-> see *privileged admission profile*

admission set

In *FTAC*, the admission set for a particular user ID defines which FT functions the user ID may use and for which *partner systems*.

admission set, privileged

-> see *privileged admission set*

AES (Advanced Encryption Standard)

The current symmetrical encryption standard, established by NIST (National Institute of Standards and Technology), based on the Rijndael algorithm, developed at the University of Leuven (B). The openFT product family uses the AES method to encrypt the request description data and possibly also the file contents.

alphanumeric

Alphanumeric characters comprise alphabetic and numeric characters, i.e. the letters A-Z and the digits 0-9 as well as the additional characters $, @, #.

AMODE

Specification for addressing a module (24-bit or 31-bit addresses).

ANSI code

Standardized 8-bit character code for message exchange. The acronym stands for "American National Standards Institute".

API (Application Program Interface)

An interface that is freely available to application programmers. It provides a set of interface mechanisms designed to support specific functionalities.

asynchronous request

Once the *FT request* has been submitted, it is processed independently of the user. The user can continue working once the system has confirmed acceptance of the request. (see also *synchronous request*).

authentication

Process used by openFT to check the unique identity of the request partner.

basic functions

Most important file transfer functions. Several basic functions are defined in the *admission set* which can be used by a login name. The six basic functions are:

- inbound receive
- inbound send
- inbound follow-up processing
- inbound file management
- outbound receive
- outbound send

central administration

Central administration in openFT incorporates the *remote administration* and *ADM traps* functions and requires the use of a *remote administration server*.

client

- Term derived from client/server architectures: the partner that makes use of the services provided by a *server*.
- Logical instance which submits requests to a *server.*

cluster

A number of computers connected over a fast network and which in many cases can be seen as a single computer externally. The objective of clustering is generally to increase the computing capacity or availability in comparison with a single computer.

Comma Separated Value (CSV)

This is a quasi-tabular output format that is very widely used in the PC environment in which the individual fields are separated by a semicolon “;”. It permits the further processing of the output from the most important openFT commands using separate tools.

communication computer

Computer for constructing a *data communication system*.

communication controller

-> see *preprocessor*

compression

This means that several identical successive characters can be reduced to one character and the number of characters is added to this. This reduces transfer times.

computer network, open

-> see *open computer network*

connectivity

In general, the ability of systems and partners to communicate with one another. Sometimes refers simply to the communication possibilities between transport systems.

cross domain connection

A connection between computers that are located in different SNA domains. A cross domain connection from a TRANSDATA network to an SNA network requires the software product TRANSIT-CD to be used as a gateway.

cross network connection

A connection between computer that are located in different SNA networks. A cross network connection from a TRANSDATA network to one or more SNA networks requires the software product TRANSIT-CD and, depending on the configuration, may also require TRANSIT-SNI to be used as a *gateway*.

data communication system

Sum of the hardware and software mechanisms which allow two or more communication partners to exchange data while adhering to specific rules.

data compression

Reducing the amount of data by means of compressed representation.

data encoding

Way in which an *FT system* represents characters internally.

Data Encryption Standard (DES)

International data encryption standard for improved security. The DES procedure is used in the FT products to encrypt the request description data and possibly the request data if connections are established to older versions of openFT that do not support *AES*.

data protection

- In the narrow sense as laid down by law, the task of protecting personal data against misuse during processing in order to prevent the disclosure or misappropriation of personal information.
- In the wider sense, the task of protecting data throughout the various stages of processing in order to prevent the disclosure or misappropriation of information relating to oneself or third parties.

data security

Technical and organizational task responsible for guaranteeing the security of data stores and data processing sequences, intended in particular to ensure that

- only authorized personnel can access the data,
- no undesired or unauthorized processing of the data is performed,
- the data is not tampered with during processing,
- the data is reproducible.

data set

File.

DHCP

Service in TCP/IP networks that automatically assigns IP addresses and TCP/IP parameters to clients on request.

Direct Access Storage Device (DASD)

Disk storage device.

directory

Directories are folders in the hierarchical file system of a Unix system (including POSIX) or a Windows system that can contain files and/or further directories. openFT for z/OS interprets, on the one hand, the contents of a PO or PDSE data set (and the members included in it) as a directory, and on the other hand also all files with a common name up to a qualifying delimiter (dot).

emulation

Components that mimic the properties of another device.

Explorer

A program from Microsoft that is supplied with Windows operating systems to facilitate navigation within the file system.

file attributes

A file’s properties, for example the size of the file, access rights to the file or the file’s record structure.

file management

Possibility of managing files in the remote system. The following actions are possible:

- Create directories
- Display and modify directories
- Delete directories
- Display and modify file attributes
- Rename files
- Delete files.

file processing

The openFT “file processing” function makes it possible to send a receive request in which the output of a remote command or program is transferred instead of a remote file.

file transfer request

-> see *FT- request*

firewall processor

Processor which connects two networks. The possible access can be controlled precisely and also logged.

fixed-length record

A record in a file all of whose records possess the same, agreed length. It is not necessary to indicate this length within the file.

follow-up processing

FT function that initiates execution of user-specified commands or statements in the *local* and/or the *remote system* after an *FT request* has been completed. The user may define different follow-up processing, depending on the success or failure of FT request processing. See also *preprocessing* and *postprocessing*.

follow-up processing request

Statements contained within an *FT request* for *follow-up processing* to be performed after file transfer.

FT administrator

Person who administers the openFT product installed on a computer, i.e. who is responsible, among other things, for the entries in the *network description file* or the *partner list* as well as for controlling resources.

FT request

Request to an *FT system* to transfer a file from a *sending system* to a *receive system* and (optionally) start *follow-up processing requests*.

FT system

System for transferring files that consists of a computer and the software required for file transfer.

FT trace

Diagnostic function that logs FT operation.

FTAC (File Transfer Access Control)

Extended access control for file transfer and file management. In the case of BS2000 and z/OS, this is implemented by means of the product openFT-AC, for other operating systems it is a component of the openFT product, e.g. in openFT for Unix systems or openFT for Windows systems.

FTAC administrator

Person who manages openFT-AC on a computer.
The FTAC administrator specifies for their system, among other things, the security-technical framework in the form of a standard admission set that is valid for all users.
In z/OS the FTAC administrator is also responsible for managing admission sets and authorization profiles.

FTAC logging function

Function which FTAC uses to log each access to the protected system via file transfer.

FTADM protocol

Protocol used for communication between two openFT instances in order to perform *remote administration* or transfer *ADM traps*.

FTAM protocol (File Transfer, Access and Management)

Protocol for file transfer standardized by the “International Organization for Standardization” (ISO) (ISO 8571, FTAM).

FTP partner

Partner system that uses *FTAM protocols* for communication.

FTP protocol

Manufacturer-independent protocol for file transfer in TCP/IP networks.

gateway

Generally understood to mean a computer that connects two or more networks and which does not function as a bridge. Variants: gateway at network level (= router or OSI relay), transport and application gateway.

gateway processor

Communication computer that links a computer network to another computer network. The mapping of the different protocols of the various computer networks takes place in gateway processors.

Generalized Trace Facility (GTF)

IBM tool for generating traces (in particular for monitoring the data traffic between an application program and the relevant VTAM applications and between VTAM applications and the data communication line).

heterogeneous network

A network consisting of multiple subnetworks functioning on the basis of different technical principles.

homogeneous network

A network constructed on the basis of a single technical principle.

identification

Procedure making it possible to identify a person or object.

IEBCOPY

IBM tool for copying libraries (PO or PDSE data sets).

IEBGENER

IBM tool for copying sequential files (PS data sets).

IEBPTPCH

IBM tool for printing files.

inbound file management

Request issued in a *remote system* for which directories or file attributes of the *local system* can be displayed, file attribute modified or local file deleted.

inbound follow-up processing

Request issued in a *remote system* with *follow-up processing* in the *local system*.

inbound receive

Request issued in the *remote system*, for which a file is received in the *local system*.

inbound request / inbound submission

Request issued in another system, i.e. for this request.

inbound send

Request issued in a *remote system* for which a file is sent from the *local system* to the remote system.

initiator

Here: *FT system* that submits an *FT request*.

instance / entity

A concept of OSI architecture: active element in a layer. Also see *openFT instance*.

instance ID

A network-wide, unique address of an openFT instance.

integrity

Unfalsified, correct data following the processing, transfer and storage phases.

Interactive Problem Control System (IPCS)

IBM tool for formatting a machine-readable (unformatted) dump.

interoperability

Capability of two *FT systems* to work together.

ISO/OSI reference model

The ISO/OSI Reference Model is a framework for the standardization of communications between open systems. (ISO=International Standards Organization).

ISPF, ISPF/PDF

Menu-driven utilities for software development and for conducting a (TSO) dialog.

job

A sequence of JCL statements (batch).

job transfer

Transfer of a file that constitutes a *job* in the *receive system* and is initiated as a job there.

library

File with internal structure (members)

library member

Part of a library. A library member may in turn be subdivided into a number of records.

Local Area Network (LAN)

Originally a high-speed network with limited physical extension. Nowadays, any network, that uses CSMA/CD, Token Ring or FDDI irrespective of the range (see also *WAN Wide Area Network*).

local system

The *FT system* at which the user is working.

logging function

Function used by openFT to log all file transfer accesses to the protected system.

log record

Contains information about access checks performed by openFT (FTAC log record) or about a file transfer or remote administration request which is started when the access check was successful (FT log record or ADM log record).

Logical Unit (LU)

Interface between an application program and the SNA data communications network. The LU type describes the communications characteristics.

Login authorization

Transfer admission to a computer which (as a rule) consists of the login name and the password, and authorizes dialog operation, see also *LOGON authorization*.

LOGON authorization

Transfer admission authorizing access to a computer. The LOGON authorization (normally) consists of user ID, account number and password and authorizes the user to make use of interactive operation.

mainframe

Computer (consisting of one or more processors) which runs under the control of a universal operating system (e.g. BS2000 or z/OS).
Synonyms: BS2000 computer, host computer.

Network Control Program (NCP)

Operating system of the front-end-processor for SNA hosts.

NetMaster

Tool for controlling a data communication system.

NetView

IBM tool for controlling a data communication system.

network description file

File used up to openFT V9 that contains specifications concerning *remote systems* (*FT systems*).

object

Passive element in a DP system that contains or receives data and which can be the object of an operation such as read, write or execute etc.
Examples: files, user IDs

open computer network

Computer network in which communication is governed by the rules of ISO/OSI. Interoperation of different computers from various vendors is made possible by defined *protocols*.

openFT instance

Several openFT systems, so-called openFT instances, can be running simultaneously on on a Sysplex cluster. Each instance has its own address (instance ID, virtual host) and is comprised of the loaded code of the openFT products (including add-on products if they are available) and of the variable files such as the network description file or partner list, logging files, key library, request queue, etc.

openFT partner

Partner system which is communicated with using *openFT protocols*.

openFT protocols

Standardized *protocols* for file transfer (SN77309, SN77312).

openFT-FTAM

Add-on product for openFT (for BS2000, Unix systems and Windows systems) that supports file transfer using FTAM protocols. FTAM stands for File Transfer, Access and Management (ISO 8571).

operating parameters

Parameters that control the *resources* (e.g. the permissible number of connections).

outbound request / outbound submission

Request issued in your own processor.

outbound receive

Request issued locally for which a file is received in the *local system*.

outbound send

Request issued locally for which a file is sent from the *local system*.

owner of an FT request

User ID in the *local system* or *remote system* under which the *FT request* is started (or submitted):

- The owner of an FT request submitted on the local system is the user ID under which the request was issued.
- The owner of an FT request submitted on a remote system is the user ID accessed for the request on the local system (TRANSFER-ADMISSION).

partitioned data set extended (PDSE data set)

Library in the IBM z/OS Data Management System. Contains individual members and can be used instead of a partitioned organized data set. The IBM software product "Data Facility Storage Management Subsystem" (DFSMS) is required to use PDSE.

partitioned organized data set (PO data set)

Library of the IBM z/OS Data Management System. Contains individual members.

partner

-> see *partner system*

partner list

File containing specifications concerning *remote systems* (*FT systems*).

partner system

Here: *FT system* that carries out *FT requests* in cooperation with the *local system*.

password

Sequence of characters that a user must enter in order to access a user ID, file, job variable, network node or application. The user ID password serves for user *authentication*. It is used for access control. The file password is used to check access rights when users access a file (or job variable). It is used for file protection purposes.

physical sequential data set / PS data set

Sequential file in the IBM z/OS Data Management System; similar to a BS2000 SAM file.

Physical Unit (PU)

Each node of an SNA network contains a Physical Unit (PU) as an addressable instance. This is responsible for monitoring the connection to the host and for monitoring the *Logical Units* (LUs).

port number

Number that uniquely identifies a TCP/IP application or the end point of a TCP/IP connection within a processor.

POSIX (Portable Open System Interface)

Board and standards laid down by it for interfaces that can be ported to different system platforms.

postprocessing

openFT makes it possible to process the received data in the receiving system through a series of operating system commands, under the process control of openFT (in contrast to *follow-up processing*).

preprocessing

The preprocessing facility in openFT can be used to send a receive request in which the outputs of a remote command or program are transferred instead of a file. This makes it possible to query a database on a remote system, for example. Preprocessing also may be issued locally.

preprocessor / communication controller

A processor system connected upstream of the mainframe which performs special communication tasks in the network. Synonym: communication processor.

private key

Secret decryption key used by the recipient to decrypt a message that was encrypted using a *public key*. Used by a variety of encryption procedures including the *RSA procedure*.

privileged admission profile

Admission profile that allows the user to exceed the *FTAC administrator's* presettings in the *admission set*. This must be approved by the *FTAC administrator* who is the only person able to privilege admission profiles.

privileged admission set

Admission set belonging to the *FTAC administrator*.

procedure

Here: command procedure, corresponds in principle to an IBM CLIST or REXX procedure.

profile

In OSI, a profile is a standard which defines which protocols may be used for any given purpose and specifies the required values of parameters and options. Here: a set of commands assigned to a user ID. The permissibility of these commands is ensured by means of syntax files. See also *admission profile*, *privileged admission profile*.

protocol

Set of rules governing information exchange between peer partners in order to achieve a defined objective. This usually consists of a definition of the messages that are to be exchanged and the correct sequencing of messages including the handling of errors and other exceptions.

public key

Public encryption key defined by the receiver of a message, and made public or made known to the sender of the message. This allows the sender to encrypt messages to be sent to the receiver. Public keys are used by various encryption methods, including the *Rivest Shamir Adleman (RSA) procedure*. The public key must match the *private key* known only to the receiver.

RACF

IBM program product for system and data access control.

receive file

File in the *receive system* in which the data from the *send file* is stored.

receive system

System to which a file is sent. This may be the *local system* or the *remote system*.

record

Set of data that is treated as a single logical unit.

relay program

Program in a *gateway processor* that maps the different protocols onto one another.

remote administration

Administration of openFT instances from remote computers.

remote administration server

Central component required for *remote administration* and for *ADM traps*. A remote administration server runs on a Unix or Windows system running openFT as of V11.0. If it is used for *remote administration*, it contains all the configuration data required for this purpose.

remote administrator

Role configured on the *remote administration server* and which grants permission to execute certain administration functions on certain openFT instances.

remote system

-> see *partner system*

request

Here: *FT request*

request queue

File containing *asynchronous requests* and their processing statuses.

request identification / request ID

The (serial) number assigned to the request by the system. In some commands, users are able to identify the request on the basis of this number.
Here: number that identifies an *FT request.*

request management

FT function responsible for managing *FT requests*; it ensures request processing from the submission of a request until its complete processing or termination.

request number

-> see *request identification*

request storage

FT function responsible for storing *FT requests* until they have been fully processed or terminated.

resources

Hardware and software components needed by the *FT system* to execute an *FT request* (, connections, lines). These resources are controlled by the *operating parameters*.

responder

Here: *FT system* addressed by the *initiator*.

restart

Automatic continuation of an *FT request* following an interruption.

restart point

Point up to which the data of the *send file* has been stored in the *receive file* when a file transfer is interrupted and at which the transfer of data is resumed following a *restart*.

result list[ing]

List with information on a completed file transfer. This is supplied to the user in the *local system* and contains information on his or her *FT requests*.

REXX

IBM procedure language.

RFC (Request for Comments)

Procedure used on the Internet for commenting on proposed standards, definitions or reports. Also used to designate a document approved in this way.

RFC1006

Supplementary protocol for the implementation of ISO transport services (transport class 0) using TCP/IP.

Rivest-Shamir-Adleman-procedure (RSA procedure)

Encryption procedure named after its inventors that operates with a key pair consisting of a *public key* and a *private key*. Used by the openFT product family in order to reliably check the identity of the partner system and to transmit the AES key to the partner system for encrypting the file contents.

Secure FTP

Method by which a connection is tunneled using the *FTP protocol*, thus allowing secure connections with encryption and *authentication*.

security level

When FTAC is used, the security level indicates the required level of protection against a *partner system*.

send file

File in the *sending system* from which data is transferred to the *receive file*.

sending system

Here: *FT system* that sends a file. This may be the *local system* or the *remote system*.

server

Logical entity or application component which executes a client's requests and assures the (coordinated) usage of all the generally available services (File, Print, data base, Communication, etc.). May itself be the client of another server.

service

- As used in the OSI architecture: a service is the set of functions that a service provider makes available at a service access point.
- As used in the client/server architecture: a set of functions that a server makes available to its clients.
- Term used in Windows: A program, routine or process used to perform a particular system function to support other programs, in particular on a low level (hardware-related).

session

- In OSI, the term used for a layer 5 connection.
- In SNA, a general term for a connection between communication partners (applications, devices or users).

session selector

Subaddress used to address a *session* application.

SMF (System Management Facility)

IBM tool for collecting accounting data and statistics.

SMP/E (System Modification Program/Extended)

IBM product used to install and manage the software products, their versions and corrections.

SNA network

Data communication system that implements the Systems Network Architecture (SNA) of IBM.

SNMP (Simple Network Management Protocol)

Protocol for TCP/IP networks defined by the Internet Engineering Task Force (IETF) for the transfer of management information.

standard admission set

This standard admission set applies by default to all users for whom there is no dedicated admission set. These default settings may be restricted further by the user for his or her own admission set.

string

Character string

SU Privilege

Privilege of an FTAC administrator in z/OS. This privilege allows the administrator to set up admission profiles for which TRANSFER-ADMISSIONS have been released on other user IDs without the need to know the current password. This privilege is defined in the FTACADM member of the parameter library.

synchronous request

The user task that submitted the *FT request* waits for transfer to terminate. The user cannot continue working (see also *asynchronous request*).

system

-> see *FT- system*

system, local

-> see *local system*

system, remote

-> see *remote system*

task

Entity responsible for executing one or more programs within a *job*.

TCP/IP (Transmission Control Protocol / Internet Protocol)

Widely used data transmission protocol (corresponds approximately to layers 3 and 4 of the *ISO/OSI reference model*, i.e. network and transport layers); originally developed for the ARPANET (computer network of the US Ministry of Defense) it has now become a de-facto standard.

TOP-SECRET

Program authored by the company Computer Associates for data and system access control.

transfer admission

Authorization for file transfer and file management when using FTAC. The transfer admissions is then used in place of the *LOGON or LOGIN authorization*.

Transmission Control Protocol / Internet Protocol

-> see *TCP/IP*

transport connection
Logical connection between two users of the transport system (terminals or applications).

transport layer
Layer 4 of the *ISO/OSI reference model* on which the data transport protocols are handled.

transport protocol
Protocol used on the *transport layer*

transport selector (T-selector)
Subaddress used to address an ISO-8072 application in the *transport layer*.

transport system

- The part of a system or architecture that performs approximately the functions of the four lower OSI layers, i.e. the transport of messages between the two partners in a communication connection.
- Sum of the hardware and software mechanisms that allow data to be transported in computer networks.

Unicode
The universal character encoding, maintained by the Unicode Consortium. This encoding standard provides the basis for processing, storage and interchange of text data in any language in all modern software and information technology protocols. The Unicode Standard defines three Unicode encoding forms: UTF-8, UTF-16 and UTF-32.

UNIX®
Registered trademark of the Open Group for a widespread multiuser operating system. A system may only bear the name UNIX if it has been certified by the Open Group.

Unix system
Commonly used designation for an operating system that implements functions typical of UNIX® and provides corresponding interfaces. POSIX and Linux are also regarded as Unix systems.

user identification / user ID
A name with a maximum length of eight characters. The user ID identifies the user when accessing the system. All files are set up under a user ID. .

variable length record

A record in a file all of whose records may be of different lengths. The record length must either be specified in a record length field at the start of the record or must be implicitly distinguishable from the next record through the use of a separator (e.g. Carriage Return - Line Feed).

VSAM

IBM file access method for sequential, direct and indexed access.

VTAM

IBM telecommunication access method.

WAN (Wide Area Network)

A public or private network that can span large distances but which runs relatively slowly and with higher error rates when compared to a *LAN*. Nowadays, these definitions have only limited validity. Example: in ATM networks.

Abbreviations

ACF/NCP	Advanced Communications Function/Network Control Program
ACF/VTAM	Advanced Communications Function/Virtual Telecommunicatio Access Method
ACF-2	Access Control Facility 2
AMODE	addressing mode
APF	Authorized Program Facility
ASCII	American Standard Code for Information Interchange
CCS	Coded Character Set
CCSN	Coded Character Set Name
CPPL	Command Processor Parameter List
CSV	Comma Separated Value
DA	Direct Access (data set)
DAS	Data Access Service
DASD	Direct Access Storage Device
DCAM	Data Communication Access Method
DES	Data Encryption Standard
DFSMS	Data Facility Storage Management Subsystem
DMS	Data Management System
DNS	Domain Name System
DSCB	Data Set Control Block
EBCDIC	Extended Binary Coded Decimal Interchange Code
FJAM	File Job Access Method
FT	File Transfer
FTAC	File Transfer Access Control
FTAM	File Transfer, Access and Management
FTP	File Transfer Protocol
GTF	Generalized Trace Facility

HSM	Hierarchical Storage Manager
IPCS	Interactive Problem Control System
ISO	International Organization for Standardization
ISPF	Interactive System Productivity Facility
ISPF/PDF	ISPF/Program Development Facility
JCL	Job Control Language
JES	Job Entry Subsystem
Kb	Kilobyte
LAN	Local Area Network
LMS	Library Maintenance System
LU	Logical Unit
Mb	Megabyte
MVS	Multiple Virtual Storage
MVS/ESA	MVS/Enterprise System Architecture
MVS/SP	MVS/System Product
MVS/XA	MVS/Extended Architecture
NCP	Network Control Program
NPSI	NCP Packet Switching Interface
NDMS	Network Data Management System
OMVS	OpenEdition MVS
OSI	Open Systems Interconnection
PDF	Program Development Facility
PDN *	Program System for Teleprocessing and Network Control
PDS	Partitioned Data Set
PDSE	Partitioned Data Set Extended
PO	Partitioned Organized (data set)
POSIX	Portable Open System Interface
PS	Physical Sequential (data set)
PSCB	Protected Step Control Block
PTF	Program Temporary Fix
PU	Physical Unit
PUT	Program Update Tape
RACF	Resource Access Control Facility

REXX	Restructured Extended Executor (language)
RFC	Request for Comments
RFC1006	Request for Comments 1006
RMODE	Residence Mode
RSA	Rivest, Shamir, Adleman
SAM	Sequential Access Method
SDSF	System Display and Search Facility
SMF	System Management Facility
SMP/E	System Modification Program/Extended
SNA	Systems Network Architecture
SSL	Secure Socket Layer
SVC	Supervisor Call
TCP/IP	Transmission Control Protocol/Internet Protocol
TSO	Time Sharing Option
TSO/E	TSO/Extension
TSS	Top-Secret
UPT	User Profile Table
VSAM	Virtual Storage Access Method
VSAM-ES	Virtual Storage Access Method - Entry Sequenced
VTAM	Virtual Telecommunication Access Method
WAN	Wide Area Network

* German abbreviation

Additional documentation

The available literature as well as current information about the openFT line of products can be found on the Internet under http://manuals.ts.fujitsu.com/
Here you will also find pdf copies of all manuals which you can download.

The appropriate documentation from IBM can be obtained on the Internet using your customer number in the usual manner.

Index